UNITED STATES AIR FORCE

SUMMER RESEARCH PROGRAM -- 1998

SUMMER FACULTY RESEARCH PROGRAM FINAL REPORTS

VOLUME 4

ROME LABORATORY

RESEARCH & DEVELOPMENT LABORATORIES

5800 Uplander Way

Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Colonel Jan Cerveny

Program Manager, RDL
Scott Licoscos

Program Administrator, RDL
Johnetta Thompson

Program Administrator, RDL
Rebecca Kelly-Clemmons

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1998

PREFACE

Reports in this volume are numbered consecutively beginning with number 1. Each report is paginated with the report number followed by consecutive page numbers, e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

This document is one of a set of 15 volumes describing the 1998 AFOSR Summer Research Program. The following volumes comprise the set:

VOLUME	TITLE
1	Program Management Report
	Summer Faculty Research Program (SFRP) Reports
2	Armstrong Laboratory
3	Phillips Laboratory
4	Rome Laboratory
5A & 5B	Wright Laboratory
6	Arnold Engineering Development Center, Air Logistics Centers, United States Air Force Academy and Wilford Hall Medical Center
	Graduate Student Research Program (GSRP) Reports
7	Armstrong Laboratory
8	Phillips Laboratory
9	Rome Laboratory
10	Wright Laboratory
11	Arnold Engineering Development Center, and Wilford Hall Medical Center
	High School Apprenticeship Program (HSAP) Reports
12	Armstrong Laboratory
13	Phillips Laboratory
14	Rome Laboratory
15A, 15B & 15C	Wright Laboratory

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-

0779

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December, 1998	

4. TITLE AND SUBTITLE
1998 Summer Research Program (SRP), Summer Faculty Research Program (SFRP), Final Reports, Volume 4, Rome Laboratory

5. FUNDING NUMBERS
F49620-93-C-0063

6. AUTHOR(S)
Gary Moore

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Research & Development Laboratories (RDL)
5800 Uplander Way
Culver City, CA 90230-6608

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research (AFOSR)
801 N. Randolph St.
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The United States Air Force Summer Research Program (USAF-SRP) is designed to introduce university, college, and technical institute faculty members, graduate students, and high school students to Air Force research. This is accomplished by the faculty members (Summer Faculty Research Program, (SFRP)), graduate students (Graduate Student Research Program (GSRP)), and high school students (High School Apprenticeship Program (HSAP)) being selected on a nationally advertised competitive basis during the summer intersession period to perform research at Air Force Research Laboratory (AFRL) Technical Directorates, Air Force Air Logistics Centers (ALC), and other AF Laboratories. This volume consists of a program overview, program management statistics, and the final technical reports from the SFRP participants at the Rome Laboratory.

14. SUBJECT TERMS
Air Force Research, Air Force, Engineering, Laboratories, Reports, Summer, Universities, Faculty, Graduate Student, High School Student

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

SRP Final Report Table of Contents

1. INTRODUCTION

The Summer Research Program (SRP), sponsored by the Air Force Office of Scientific Research (AFOSR), offers paid opportunities for university faculty, graduate students, and high school students to conduct research in U.S. Air Force research laboratories nationwide during the summer.

Introduced by AFOSR in 1978, this innovative program is based on the concept of teaming academic researchers with Air Force scientists in the same disciplines using laboratory facilities and equipment not often available at associates' institutions.

The Summer Faculty Research Program (SFRP) is open annually to approximately 150 faculty members with at least two years of teaching and/or research experience in accredited U.S. colleges, universities, or technical institutions. SFRP associates must be either U.S. citizens or permanent residents.

The Graduate Student Research Program (GSRP) is open annually to approximately 100 graduate students holding a bachelor's or a master's degree; GSRP associates must be U.S. citizens enrolled full time at an accredited institution.

The High School Apprentice Program (HSAP) annually selects about 125 high school students located within a twenty mile commuting distance of participating Air Force laboratories.

AFOSR also offers its research associates an opportunity, under the Summer Research Extension Program (SREP), to continue their AFOSR-sponsored research at their home institutions through the award of research grants. In 1994 the maximum amount of each grant was increased from $20,000 to $25,000, and the number of AFOSR-sponsored grants decreased from 75 to 60. A separate annual report is compiled on the SREP.

The numbers of projected summer research participants in each of the three categories and SREP "grants" are usually increased through direct sponsorship by participating laboratories.

AFOSR's SRP has well served its objectives of building critical links between Air Force research laboratories and the academic community, opening avenues of communications and forging new research relationships between Air Force and academic technical experts in areas of national interest, and strengthening the nation's efforts to sustain careers in science and engineering. The success of the SRP can be gauged from its growth from inception (see Table 1) and from the favorable responses the 1997 participants expressed in end-of-tour SRP evaluations (Appendix B).

AFOSR contracts for administration of the SRP by civilian contractors. The contract was first awarded to Research & Development Laboratories (RDL) in September 1990. After completion of the 1990 contract, RDL (in 1993) won the recompetition for the basic year and four 1-year options.

2. PARTICIPATION IN THE SUMMER RESEARCH PROGRAM

The SRP began with faculty associates in 1979; graduate students were added in 1982 and high school students in 1986. The following table shows the number of associates in the program each year.

YEAR	SRP Participation, by Year			TOTAL
	SFRP	GSRP	HSAP	
1979	70			70
1980	87			87
1981	87			87
1982	91	17		108
1983	101	53		154
1984	152	84		236
1985	154	92		246
1986	158	100	42	300
1987	159	101	73	333
1988	153	107	101	361
1989	168	102	103	373
1990	165	121	132	418
1991	170	142	132	444
1992	185	121	159	464
1993	187	117	136	440
1994	192	117	133	442
1995	190	115	137	442
1996	188	109	138	435
1997	148	98	140	427
1998	85	40	88	213

Beginning in 1993, due to budget cuts, some of the laboratories weren't able to afford to fund as many associates as in previous years. Since then, the number of funded positions has remained fairly constant at a slightly lower level.

3. RECRUITING AND SELECTION

The SRP is conducted on a nationally advertised and competitive-selection basis. The advertising for faculty and graduate students consisted primarily of the mailing of 8,000 52-page SRP brochures to chairpersons of departments relevant to AFOSR research and to administrators of grants in accredited universities, colleges, and technical institutions. Historically Black Colleges and Universities (HBCUs) and Minority Institutions (MIs) were included. Brochures also went to all participating USAF laboratories, the previous year's participants, and numerous individual requesters (over 1000 annually).

RDL placed advertisements in the following publications: *Black Issues in Higher Education, Winds of Change,* and *IEEE Spectrum.* Because no participants list either *Physics Today* or *Chemical & Engineering News* as being their source of learning about the program for the past several years, advertisements in these magazines were dropped, and the funds were used to cover increases in brochure printing costs.

High school applicants can participate only in laboratories located no more than 20 miles from their residence. Tailored brochures on the HSAP were sent to the head counselors of 180 high schools in the vicinity of participating laboratories, with instructions for publicizing the program in their schools. High school students selected to serve at Wright Laboratory's Armament Directorate (Eglin Air Force Base, Florida) serve eleven weeks as opposed to the eight weeks normally worked by high school students at all other participating laboratories.

Each SFRP or GSRP applicant is given a first, second, and third choice of laboratory. High school students who have more than one laboratory or directorate near their homes are also given first, second, and third choices.

Laboratories make their selections and prioritize their nominees. AFOSR then determines the number to be funded at each laboratory and approves laboratories' selections.

Subsequently, laboratories use their own funds to sponsor additional candidates. Some selectees do not accept the appointment, so alternate candidates are chosen. This multi-step selection procedure results in some candidates being notified of their acceptance after scheduled deadlines. The total applicants and participants for 1998 are shown in this table.

1998 Applicants and Participants			
PARTICIPANT CATEGORY	TOTAL APPLICANTS	SELECTEES	DECLINING SELECTEES
SFRP	382	85	13
(HBCU/MI)	(0)	(0)	(0)
GSRP	130	40	7
(HBCU/MI)	(0)	(0)	(0)
HSAP	328	88	22
TOTAL	840	213	42

4. SITE VISITS

During June and July of 1998, representatives of both AFOSR/NI and RDL visited each participating laboratory to provide briefings, answer questions, and resolve problems for both laboratory personnel and participants. The objective was to ensure that the SRP would be as constructive as possible for all participants. Both SRP participants and RDL representatives found these visits beneficial. At many of the laboratories, this was the only opportunity for all participants to meet at one time to share their experiences and exchange ideas.

5. HISTORICALLY BLACK COLLEGES AND UNIVERSITIES AND MINORITY INSTITUTIONS (HBCU/MIs)

Before 1993, an RDL program representative visited from seven to ten different HBCU/MIs annually to promote interest in the SRP among the faculty and graduate students. These efforts were marginally effective, yielding a doubling of HBCI/MI applicants. In an effort to achieve AFOSR's goal of 10% of all applicants and selectees being HBCU/MI qualified, the RDL team decided to try other avenues of approach to increase the number of qualified applicants. Through the combined efforts of the AFOSR Program Office at Bolling AFB and RDL, two very active minority groups were found, HACU (Hispanic American Colleges and Universities) and AISES (American Indian Science and Engineering Society). RDL is in communication with representatives of each of these organizations on a monthly basis to keep up with the their activities and special events. Both organizations have widely-distributed magazines/quarterlies in which RDL placed ads.

Since 1994 the number of both SFRP and GSRP HBCU/MI applicants and participants has increased ten-fold, from about two dozen SFRP applicants and a half dozen selectees to over 100 applicants and two dozen selectees, and a half-dozen GSRP applicants and two or three selectees to 18 applicants and 7 or 8 selectees. Since 1993, the SFRP had a two-fold applicant increase and a two-fold selectee increase. Since 1993, the GSRP had a three-fold applicant increase and a three to four-fold increase in selectees.

In addition to RDL's special recruiting efforts, AFOSR attempts each year to obtain additional funding or use leftover funding from cancellations the past year to fund HBCU/MI associates.

SRP HBCU/MI Participation, By Year				
YEAR	SFRP		GSRP	
	Applicants	Participants	Applicants	Participants
1985	76	23	15	11
1986	70	18	20	10
1987	82	32	32	10
1988	53	17	23	14
1989	39	15	13	4
1990	43	14	17	3
1991	42	13	8	5
1992	70	13	9	5
1993	60	13	6	2
1994	90	16	11	6
1995	90	21	20	8
1996	119	27	18	7

6. SRP FUNDING SOURCES

Funding sources for the 1998 SRP were the AFOSR-provided slots for the basic contract and laboratory funds. Funding sources by category for the 1998 SRP selected participants are shown here.

1998 SRP FUNDING CATEGORY	SFRP	GSRP	HSAP
AFOSR Basic Allocation Funds	67	38	75
USAF Laboratory Funds	17	2	13
Slots Added by AFOSR (Leftover Funds)	0	0	0
HBCU/MI By AFOSR (Using Procured Addn'l Funds)	0	0	N/A
TOTAL	84	40	88

7. COMPENSATION FOR PARTICIPANTS

Compensation for SRP participants, per five-day work week, is shown in this table.

1998 SRP Associate Compensation

PARTICIPANT CATEGORY	1991	1992	1993	1994	1995	1996	1997	1998
Faculty Members	$690	$718	$740	$740	$740	$770	$770	$793
Graduate Student (Master's Degree)	$425	$442	$455	$455	$455	$470	$470	$484
Graduate Student (Bachelor's Degree)	$365	$380	$391	$391	$391	$400	$400	$412
High School Student (First Year)	$200	$200	$200	$200	$200	$200	$200	$200
High School Student (Subsequent Years)	$240	$240	$240	$240	$240	$240	$240	$240

The program also offered associates whose homes were more than 50 miles from the laboratory an expense allowance (seven days per week) of $52/day for faculty and $41/day for graduate students. Transportation to the laboratory at the beginning of their tour and back to their home destinations at the end was also reimbursed for these participants. Of the combined SFRP and GSRP associates, 65 % claimed travel reimbursements at an average round-trip cost of $730.

Faculty members were encouraged to visit their laboratories before their summer tour began. All costs of these orientation visits were reimbursed. Forty-three percent (85 out of 188) of faculty associates took orientation trips at an average cost of $449. By contrast, in 1993, 58 % of SFRP associates elected to take an orientation visits at an average cost of $685; that was the highest percentage of

associates opting to take an orientation trip since RDL has administered the SRP, and the highest average cost of an orientation trip.

Program participants submitted biweekly vouchers countersigned by their laboratory research focal point, and RDL issued paychecks so as to arrive in associates' hands two weeks later.

This is the third year of using direct deposit for the SFRP and GSRP associates. The process went much more smoothly with respect to obtaining required information from the associates, about 15% of the associates' information needed clarification in order for direct deposit to properly function as opposed to 7% from last year. The remaining associates received their stipend and expense payments via checks sent in the US mail.

HSAP program participants were considered actual RDL employees, and their respective state and federal income tax and Social Security were withheld from their paychecks. By the nature of their independent research, SFRP and GSRP program participants were considered to be consultants or independent contractors. As such, SFRP and GSRP associates were responsible for their own income taxes, Social Security, and insurance.

8. CONTENTS OF THE 1998 REPORT

The complete set of reports for the 1998 SRP includes this program management report (Volume 1) augmented by fifteen volumes of final research reports by the 1998 associates, as indicated below:

1998 SRP Final Report Volume Assignments

LABORATORY	SFRP	GSRP	HSAP
Armstrong	2	7	12
Phillips	3	8	13
Rome	4	9	14
Wright	5A, 5B	10	15
AEDC, ALCs, USAFA, WHMC	6	11	

APPENDIX A -- PROGRAM STATISTICAL SUMMARY

A. Colleges/Universities Represented

Selected SFRP associates represented 169 different colleges, universities, and institutions, GSRP associates represented 95 different colleges, universities, and institutions.

B. States Represented

SFRP -Applicants came from 47 states plus Washington D.C. Selectees represent 44 states.

GSRP - Applicants came from 44 states. Selectees represent 32 states.

HSAP - Applicants came from thirteen states. Selectees represent nine states.

Total Number of Participants	
SFRP	85
GSRP	40
HSAP	88
TOTAL	213

Degrees Represented			
	SFRP	GSRP	TOTAL
Doctoral	83	0	83
Master's	1	3	4
Bachelor's	0	22	22
TOTAL	186	25	109

SFRP Academic Titles	
Assistant Professor	36
Associate Professor	34
Professor	15
Instructor	0
Chairman	0
Visiting Professor	0
Visiting Assoc. Prof.	0
Research Associate	0
TOTAL	85

Source of Learning About the SRP		
Category	Applicants	Selectees
Applied/participated in prior years	177	47
Colleague familiar with SRP	104	24
Brochure mailed to institution	101	21
Contact with Air Force laboratory	101	39
IEEE Spectrum	12	1
BIIHE	4	0
Other source	117	30
TOTAL	616	162

APPENDIX B – SRP EVALUATION RESPONSES

1. OVERVIEW

Evaluations were completed and returned to RDL by four groups at the completion of the SRP. The number of respondents in each group is shown below.

Table B-1. Total SRP Evaluations Received

Evaluation Group	Responses
SFRP & GSRPs	100
HSAPs	75
USAF Laboratory Focal Points	84
USAF Laboratory HSAP Mentors	6

All groups indicate unanimous enthusiasm for the SRP experience.

The summarized recommendations for program improvement from both associates and laboratory personnel are listed below:

A. Better preparation on the labs' part prior to associates' arrival (i.e., office space, computer assets, clearly defined scope of work).

B. Faculty Associates suggest higher stipends for SFRP associates.

C. Both HSAP Air Force laboratory mentors and associates would like the summer tour extended from the current 8 weeks to either 10 or 11 weeks: the groups state it takes 4-6 weeks just to get high school students up-to-speed on what's going on at laboratory. (Note: this same argument was used to raise the faculty and graduate student participation time a few years ago.)

2. 1998 USAF LABORATORY FOCAL POINT (LFP) EVALUATION RESPONSES

The summarized results listed below are from the 84 LFP evaluations received.

1. LFP evaluations received and associate preferences:

Table B-2. Air Force LFP Evaluation Responses (By Type)

		How Many Associates Would You Prefer To Get ? (% Response)											
		SFRP				GSRP (w/Univ Professor)				GSRP (w/o Univ Professor)			
Lab	Evals Recv'd	0	1	2	3+	0	1	2	3+	0	1	2	3+
AEDC	0	-	-	-	-	-	-	-	-	-	-	-	-
WHMC	0	-	-	-	-	-	-	-	-	-	-	-	-
AL	7	28	28	28	14	54	14	28	0	86	0	14	0
USAFA	1	0	100	0	0	100	0	0	0	0	100	0	0
PL	25	40	40	16	4	88	12	0	0	84	12	4	0
RL	5	60	40	0	0	80	10	0	0	100	0	0	0
WL	46	30	43	20	6	78	17	4	0	93	4	2	0
Total	**84**	**32%**	**50%**	**13%**	**5%**	**80%**	**11%**	**6%**	**0%**	**73%**	**23%**	**4%**	**0%**

LFP Evaluation Summary. The summarized responses, by laboratory, are listed on the following page. LFPs were asked to rate the following questions on a scale from 1 (below average) to 5 (above average).

2. LFPs involved in SRP associate application evaluation process:
    a. Time available for evaluation of applications:
    b. Adequacy of applications for selection process:
3. Value of orientation trips:
4. Length of research tour:
5. a. Benefits of associate's work to laboratory:
    b. Benefits of associate's work to Air Force:
6. a. Enhancement of research qualifications for LFP and staff:
    b. Enhancement of research qualifications for SFRP associate:
    c. Enhancement of research qualifications for GSRP associate:
7. a. Enhancement of knowledge for LFP and staff:
    b. Enhancement of knowledge for SFRP associate:
    c. Enhancement of knowledge for GSRP associate:
8. Value of Air Force and university links:
9. Potential for future collaboration:
10. a. Your working relationship with SFRP:
    b. Your working relationship with GSRP:
11. Expenditure of your time worthwhile:

(Continued on next page)

12. Quality of program literature for associate:
13. a. Quality of RDL's communications with you:
    b. Quality of RDL's communications with associates:
14. Overall assessment of SRP:

Table B-3. Laboratory Focal Point Reponses to above questions

	AEDC	*AL*	*USAFA*	*PL*	*RL*	*WHMC*	*WL*
# Evals Recv'd	0	7	1	14	5	0	46
Question #							
2	-	86 %	0 %	88 %	80 %	-	85 %
2a	-	4.3	n/a	3.8	4.0	-	3.6
2b	-	4.0	n/a	3.9	4.5	-	4.1
3	-	4.5	n/a	4.3	4.3	-	3.7
4	-	4.1	4.0	4.1	4.2	-	3.9
5a	-	4.3	5.0	4.3	4.6	-	4.4
5b	-	4.5	n/a	4.2	4.6	-	4.3
6a	-	4.5	5.0	4.0	4.4	-	4.3
6b	-	4.3	n/a	4.1	5.0	-	4.4
6c	-	3.7	5.0	3.5	5.0	-	4.3
7a	-	4.7	5.0	4.0	4.4	-	4.3
7b	-	4.3	n/a	4.2	5.0	-	4.4
7c	-	4.0	5.0	3.9	5.0	-	4.3
8	-	4.6	4.0	4.5	4.6	-	4.3
9	-	4.9	5.0	4.4	4.8	-	4.2
10a	-	5.0	n/a	4.6	4.6	-	4.6
10b	-	4.7	5.0	3.9	5.0	-	4.4
11	-	4.6	5.0	4.4	4.8	-	4.4
12	-	4.0	4.0	4.0	4.2	-	3.8
13a	-	3.2	4.0	3.5	3.8	-	3.4
13b	-	3.4	4.0	3.6	4.5	-	3.6
14	-	4.4	5.0	4.4	4.8	-	4.4

3. 1998 SFRP & GSRP EVALUATION RESPONSES

The summarized results listed below are from the 120 SFRP/GSRP evaluations received.

Associates were asked to rate the following questions on a scale from 1 (below average) to 5 (above average) - by Air Force base results and over-all results of the 1998 evaluations are listed after the questions.

1. The match between the laboratories research and your field:
2. Your working relationship with your LFP:
3. Enhancement of your academic qualifications:
4. Enhancement of your research qualifications:
5. Lab readiness for you: LFP, task, plan:
6. Lab readiness for you: equipment, supplies, facilities:
7. Lab resources:
8. Lab research and administrative support:
9. Adequacy of brochure and associate handbook:
10. RDL communications with you:
11. Overall payment procedures:
12. Overall assessment of the SRP:
13. a. Would you apply again?
    b. Will you continue this or related research?
14. Was length of your tour satisfactory?
15. Percentage of associates who experienced difficulties in finding housing:
16. Where did you stay during your SRP tour?
    a. At Home:
    b. With Friend:
    c. On Local Economy:
    d. Base Quarters:
17. Value of orientation visit:
    a. Essential:
    b. Convenient:
    c. Not Worth Cost:
    d. Not Used:

SFRP and GSRP associate's responses are listed in tabular format on the following page.

Table B-4. 1997 SFRP & GSRP Associate Responses to SRP Evaluation

	Arnold	Brooks	Edwards	Eglin	Griffis	Hanscom	Kelly	Kirtland	Lackland	Robins	Tyndall	WPAFB	average
# res	6	48	6	14	31	19	3	32	1	2	10	85	257
1	4.8	4.4	4.6	4.7	4.4	4.9	4.6	4.6	5.0	5.0	4.0	4.7	4.6
2	5.0	4.6	4.1	4.9	4.7	4.7	5.0	4.7	5.0	5.0	4.6	4.8	4.7
3	4.5	4.4	4.0	4.6	4.3	4.2	4.3	4.4	5.0	5.0	4.5	4.3	4.4
4	4.3	4.5	3.8	4.6	4.4	4.4	4.3	4.6	5.0	4.0	4.4	4.5	4.5
5	4.5	4.3	3.3	4.8	4.4	4.5	4.3	4.2	5.0	5.0	3.9	4.4	4.4
6	4.3	4.3	3.7	4.7	4.4	4.5	4.0	3.8	5.0	5.0	3.8	4.2	4.2
7	4.5	4.4	4.2	4.8	4.5	4.3	4.3	4.1	5.0	5.0	4.3	4.3	4.4
8	4.5	4.6	3.0	4.9	4.4	4.3	4.3	4.5	5.0	5.0	4.7	4.5	4.5
9	4.7	4.5	4.7	4.5	4.3	4.5	4.7	4.3	5.0	5.0	4.1	4.5	4.5
10	4.2	4.4	4.7	4.4	4.1	4.1	4.0	4.2	5.0	4.5	3.6	4.4	4.3
11	3.8	4.1	4.5	4.0	3.9	4.1	4.0	4.0	3.0	4.0	3.7	4.0	4.0
12	5.7	4.7	4.3	4.9	4.5	4.9	4.7	4.6	5.0	4.5	4.6	4.5	4.6
Numbers below are percentages													
13a	83	90	83	93	87	75	100	81	100	100	100	86	87
13b	100	89	83	100	94	98	100	94	100	100	100	94	93
14	83	96	100	90	87	80	100	92	100	100	70	84	88
15	17	6	0	33	20	76	33	25	0	100	20	8	39
16a	-	26	17	9	38	23	33	4	-	-	-	30	
16b	100	33	-	40	-	8	-	-	-	-	36	2	
16c	-	41	83	40	62	69	67	96	100	100	64	68	
16d	-	-	-	-	-	-	-	-	-	-	-	0	
17a	-	33	100	17	50	14	67	39	-	50	40	31	35
17b	-	21	-	17	10	14	-	24	-	50	20	16	16
17c	-	-	-	-	10	7	-	-	-	-	-	2	3
17d	100	46	-	66	30	69	33	37	100	-	40	51	46

4. 1998 USAF LABORATORY HSAP MENTOR EVALUATION RESPONSES

Not enough evaluations received (5 total) from Mentors to do useful summary.

5. 1998 HSAP EVALUATION RESPONSES

The summarized results listed below are from the 23 HSAP evaluations received.

HSAP apprentices were asked to rate the following questions on a scale from 1 (below average) to 5 (above average)

1. Your influence on selection of topic/type of work.
2. Working relationship with mentor, other lab scientists.
3. Enhancement of your academic qualifications.
4. Technically challenging work.
5. Lab readiness for you: mentor, task, work plan, equipment.
6. Influence on your career.
7. Increased interest in math/science.
8. Lab research & administrative support.
9. Adequacy of RDL's Apprentice Handbook and administrative materials.
10. Responsiveness of RDL communications.
11. Overall payment procedures.
12. Overall assessment of SRP value to you.
13. Would you apply again next year? Yes (92 %)
14. Will you pursue future studies related to this research? Yes (68 %)
15. Was Tour length satisfactory? Yes (82 %)

	Arnold	Brooks	Edwards	Eglin	Griffiss	Hanscom	Kirtland	Tyndall	WPAFB	Totals
# resp	5	19	7	15	13	2	7	5	40	113
1	2.8	3.3	3.4	3.5	3.4	4.0	3.2	3.6	3.6	3.4
2	4.4	4.6	4.5	4.8	4.6	4.0	4.4	4.0	4.6	4.6
3	4.0	4.2	4.1	4.3	4.5	5.0	4.3	4.6	4.4	4.4
4	3.6	3.9	4.0	4.5	4.2	5.0	4.6	3.8	4.3	4.2
5	4.4	4.1	3.7	4.5	4.1	3.0	3.9	3.6	3.9	4.0
6	3.2	3.6	3.6	4.1	3.8	5.0	3.3	3.8	3.6	3.7
7	2.8	4.1	4.0	3.9	3.9	5.0	3.6	4.0	4.0	3.9
8	3.8	4.1	4.0	4.3	4.0	4.0	4.3	3.8	4.3	4.2
9	4.4	3.6	4.1	4.1	3.5	4.0	3.9	4.0	3.7	3.8
10	4.0	3.8	4.1	3.7	4.1	4.0	3.9	2.4	3.8	3.8
11	4.2	4.2	3.7	3.9	3.8	3.0	3.7	2.6	3.7	3.8
12	4.0	4.5	4.9	4.6	4.6	5.0	4.6	4.2	4.3	4.5
Numbers below are percentages										
13	60%	95%	100%	100%	85%	100%	100%	100%	90%	92%
14	20%	80%	71%	80%	54%	100%	71%	80%	65%	68%
15	100%	70%	71%	100%	100%	50%	86%	60%	80%	82%

DESIGN OF A MICROWAVE-TO-OPTICAL LINK AMPLIFIER FOR RADAR APPLICATIONS

Ercument Arvas
Professor
Department of Electrical Engineering and Computer Science

Syracuse University
121 Link Hall
Syracuse, NY 13244

Final Report for:
Summer Faculty Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
And
Rome Laboratory

July 1998

DESIGN OF A MICROWAVE-TO-OPTICAL LINK AMPLIFIER FOR RADAR APPLICATIONS

Ercument Arvas
Professor
Department of Electrical Engineering and Computer Science
Syracuse University

Abstract

The ultimate goal is to design, simulate, build and test a MMIC low noise amplifier that will perform as a link between a receiving antenna of a radar and an optical fiber that acts as the feed line to the receiver. To investigate potential problems with MMIC, this short summer period was used to first implement the device using hybrid MIC technology: Here a hybrid microwave amplifier operating in the frequency band 3.1 GHz - 3.5 GHz is designed to deliver power to a laser diode which has an equivalent input impedance of Z_{in}=4.4+ j 19 Ω in the band of interest. The design specifications include reasonably high and flat gain, low noise figure and large dynamic range. The design was simulated on microstrips using commercial software Serenade v.7.5. Detailed analysis on stability, ac performance, dc bias conditions and optimization was carried out. Finally a statistical analysis was performed to investigate the overall performance of the device. The simulation results are promising. The circuit was then built as a two-stage amplifier and preliminary test results are satisfactory.

DESIGN OF A MICROWAVE-TO-OPTICAL LINK AMPLIFIER FOR RADAR APPLICATIONS

Ercument Arvas

1.INTRODUCTION

Airborne microwave radar and communications systems of the future, both military and commercial, will require higher performance receiver front ends. This means lower system noise figures with higher versatility in adaptively forming one or multiple beams. Generally, the antenna will be mounted on the fuselage and the receiver will be mounted somewhere significantly remote from the antenna. This is not consistent with a low noise figure system, unless there is a low noise amplifier at the antenna that sets the system noise figure. In order for the system to be able to have a reasonable amount of adaptively there will have to be multiple receiver channels. To achieve multiple channels there will have to be multiple feed lines. Current technology implies bulky and heavy coaxial cables. One solution is to use optical fibers as the feed lines for each receiver channel. This will also help reduce weight and save space. The technology challenge is to match the low noise figure amplifier output impedance to the laser diode input impedance while maintaining the microwave performance of the system (i.e., dynamic range, gain etc.).

Furthermore, the low noise amplifiers could be built using MMIC technology. The use of MMIC amplifiers as the low noise amplifiers will help reduce the weight of the antenna system further, and will aid in keeping the volume of the antenna system to a level that will allow the use of these microwave systems on Unmanned Air Vehicles (UAV). Specifically this has potential as a pervasive technology enabling some system concepts for bistatic radar.

Therefore, the ultimate goal of this project is to design, simulate, build and test a low noise MMIC amplifier that will feed the received signal from the radar antenna to a laser diode exciting an optical fiber that feeds the receiver. To investigate many potential

problems expected in the MMIC design, the amplifier is first built using hybrid technology. During the summer months a detailed simulation of the hybrid amplifier was carried out using the commercial software Serenade v.7.5. Then the two-stage amplifier was built using microstrip technology. The simulation results are included and look very good. Preliminary test results look promising. The following sections give the details of the design and simulation procedure together with computed results.

2. AC DESIGN AND SIMULATION RESULTS

2.1. Design Specifications

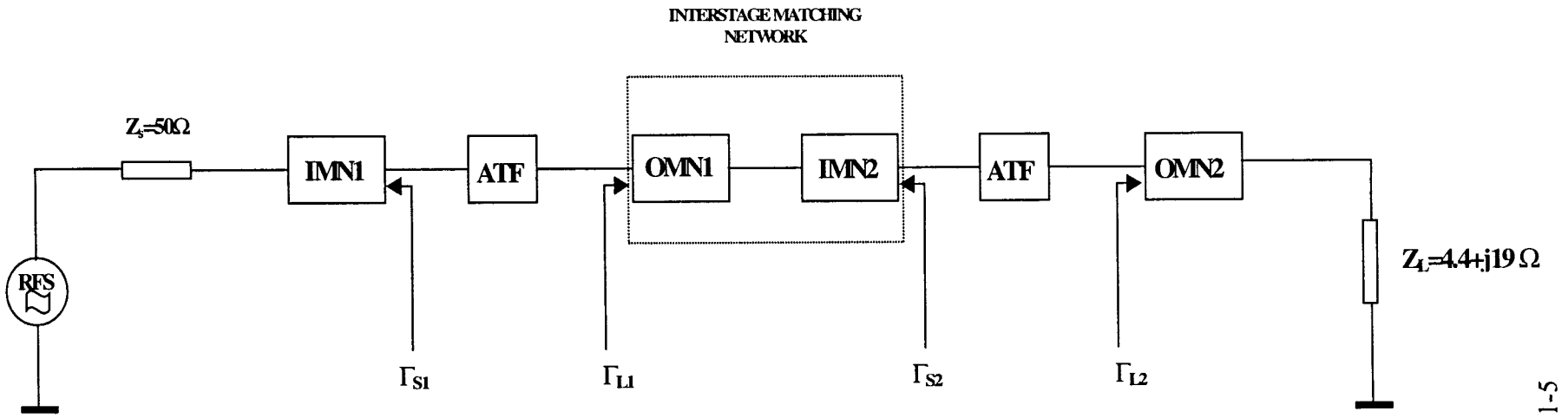
Frequency Band	: 3.1 GHz- 3.5 GHz
Gain	: 21 dB
Z_s	: 50 Ω
Z_L	: 4.4 + j 19
Method	: Distributed element hybrid
Transistor	: ATF-21186 Field Effect Transistor
Print Material	: Rogers 3003

Parameter	Symbol	Value
Dielectric constant	ε	3.0
Dielectric height	h	30 mil
Copper height	t	1.4 mil
Surface resistivity	R	10^7 Ω/sqm
Loss tangent	δ	0.0013

2.2. Design Approach

- First design a single stage for 10.5 dB gain (See references [1], [3])
- Cascade two amplifiers
- Optimize the overall circuit for Z_L= 4.4 + j 19
- Add the bias insertion network which uses quarter-wavelengths
- Re-optimize the overall circuit to realize the required gain without touching the dc bias distributed elements

2.3. Block Diagram Representation of the Two-Stage Amplifier



1-5

Fig.1. Block diagram representation of the two-stage amplifier

IMN1	: First stage input matching network
ATF	: ATF-21186 FET transistor
OMN1	: First stage output matching network
IMN2	: Second stage input matching network
OMN2	: Second stage output matching network
Γ_{S1}	: Source reflection coefficient of the first transistor
Γ_{L1}	: Load reflection coefficient of the first transistor
Γ_{S2}	: Source reflection coefficient of the second transistor
Γ_{L2}	: Load reflection coefficient of the second transistor

2.4. Two-stage Amplifier Showing the Details of the Matching Network

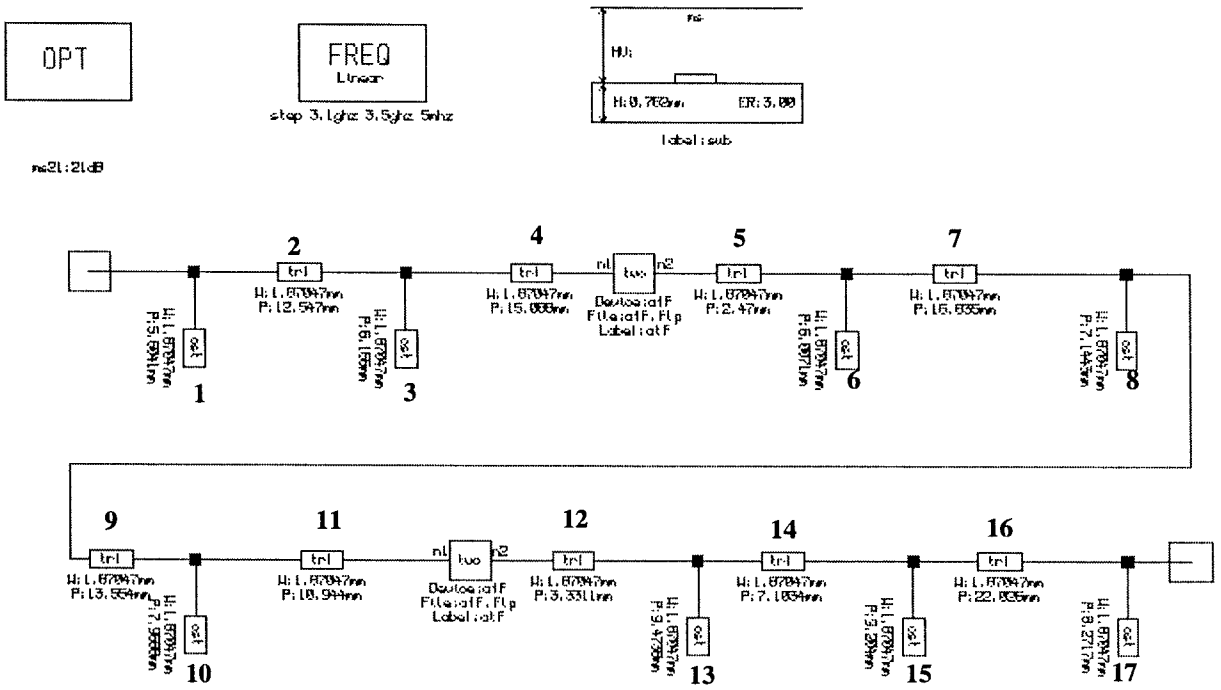


Fig.2. The two-stage amplifier with details of matching network

2.5. Physical Dimensions of the Two-stage Amplifier in Tabulated Form

Component values are given beginning from port 1. All lines are chosen to have characteristic impedance of 50 Ω. This fixes the widths of the microstrips at f=3.3 GHz to 1.87047 mm.

Table 1

Component #	Description	Width (mm)	Length (mm)
1	Stub	1.87047	5.6041
2	Series line	1.87047	12.547
3	Stub	1.87047	6.166
4	Series line	1.87047	15.088
5	Series line	1.87047	2.47
6	Stub	1.87047	6.0071
7	Series line	1.87047	16.835
8	Stub	1.87047	7.1443
9	Series line	1.87047	13.554
10	Stub	1.87047	7.9688
11	Series line	1.87047	10.944
12	Series line	1.87047	3.3311
13	Stub	1.87047	9.4738
14	Series line	1.87047	7.1034
15	Stub	1.87047	9.204
16	Series line	1.87047	22.025
17	Stub	1.87047	8.2717

The following figures show the scattering parameters of the two-port network. As it can be seen from figure 3, gain is reasonably flat in the band of interest.

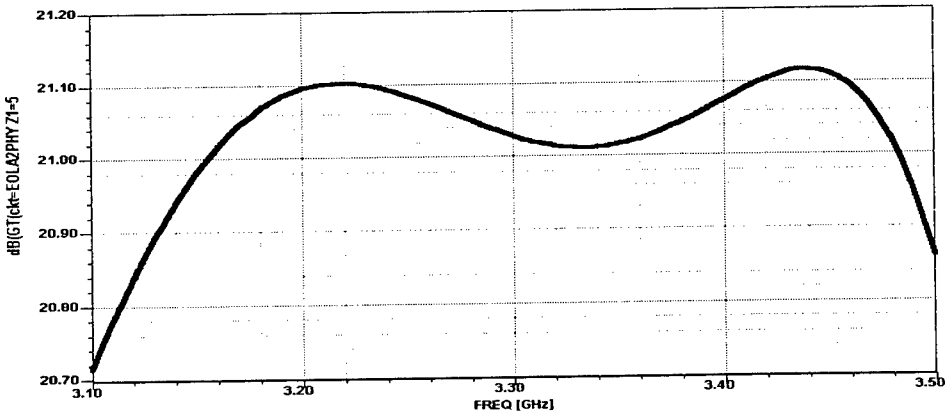


Fig.3. Transducer power gain of the two port

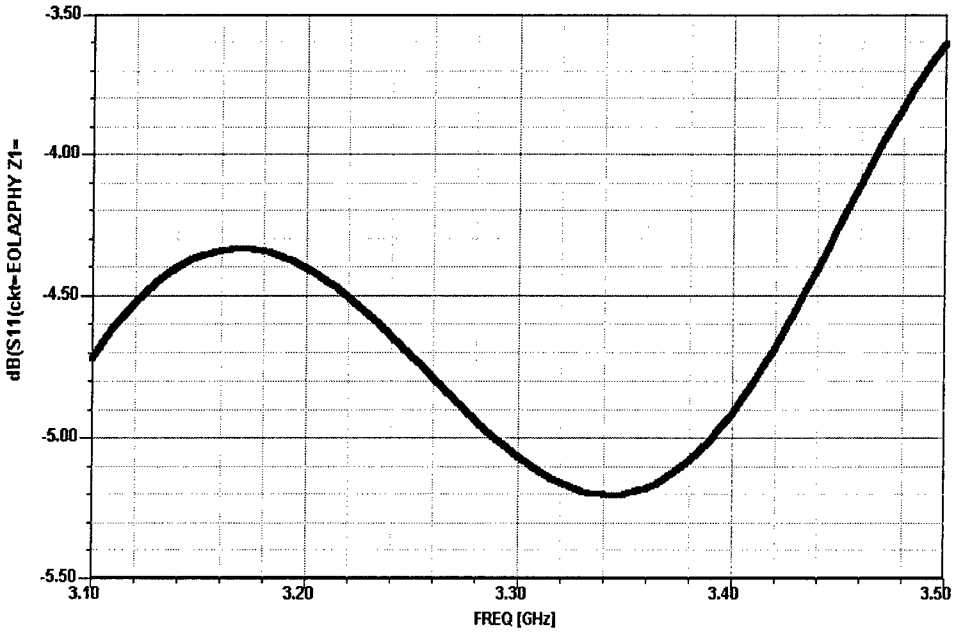


Fig.4. S11 of the two-port network

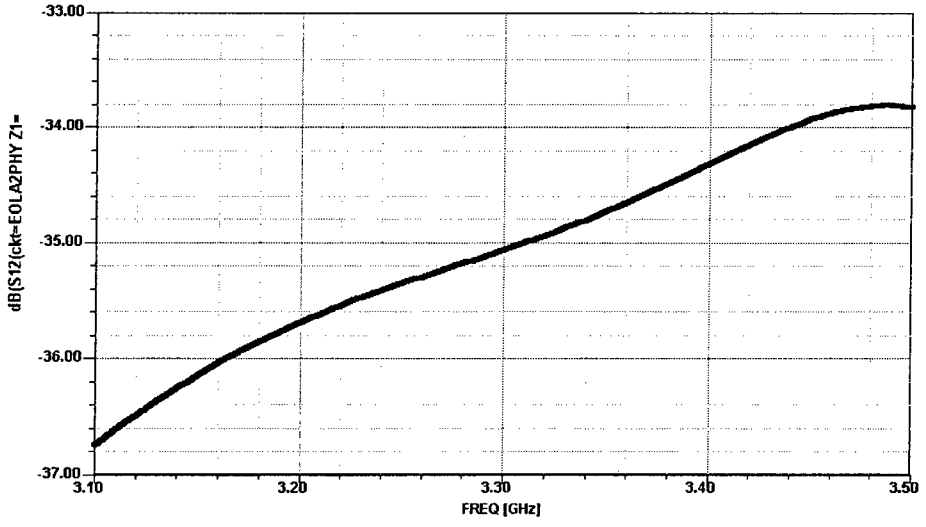


Fig.5. S12 of the two-port network

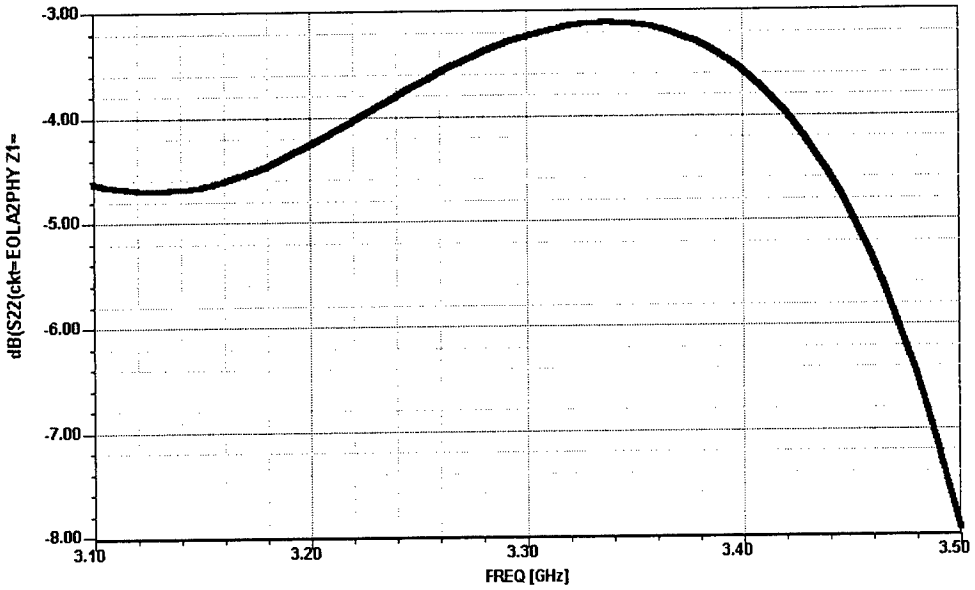


Fig.6. S22 of the two-port network

3. DC BIAS INSERTION NETWORK AND AC+DC SIMULATION RESULTS

3.1. Amplifier Bias Insertion Network

The purpose of the bias insertion network is to supply the gate and drain with the required DC voltages and currents. This network should also prevent the power supply from the RF signals of the power amplifier. While doing this, the input and output matching networks should not be affected in the frequency range of operation.

Figure 7 shows two quarter-wave transmission lines that are grounded through a capacitor between the two. A capacitor with a series resonant near the transmission band is chosen. A ferroperm HQ-0805 5% capacitor with C=10 pF and series resonant f_s=1.9Ghz. The dielectric substrate that we chose is copper (R03003) and has a dielectric constant ε_r=3.00.

The microstrip lines are connected to the main lines at gate and drain. Ideally the RF signal sees an infinite impedance, when looking into the open stub, and therefore is

blocked. This assumption is only valid at the design frequency. In order to make the network broadbanded the widths W_{d1}, W_{d2}, W_{g1}, W_{g2} of the transmission lines are different; one narrow and one wide which corresponds to a large and a small characteristic impedance.

In the DC bias network of the power amplifier, a drain source voltage of 3V is used. In the transistor data sheet, some S-parameters are given under the bias conditions V_{ds}= 3V and I_d= 70 mA. Therefore we decided to bias our transistor with V_{ds}= 3V.

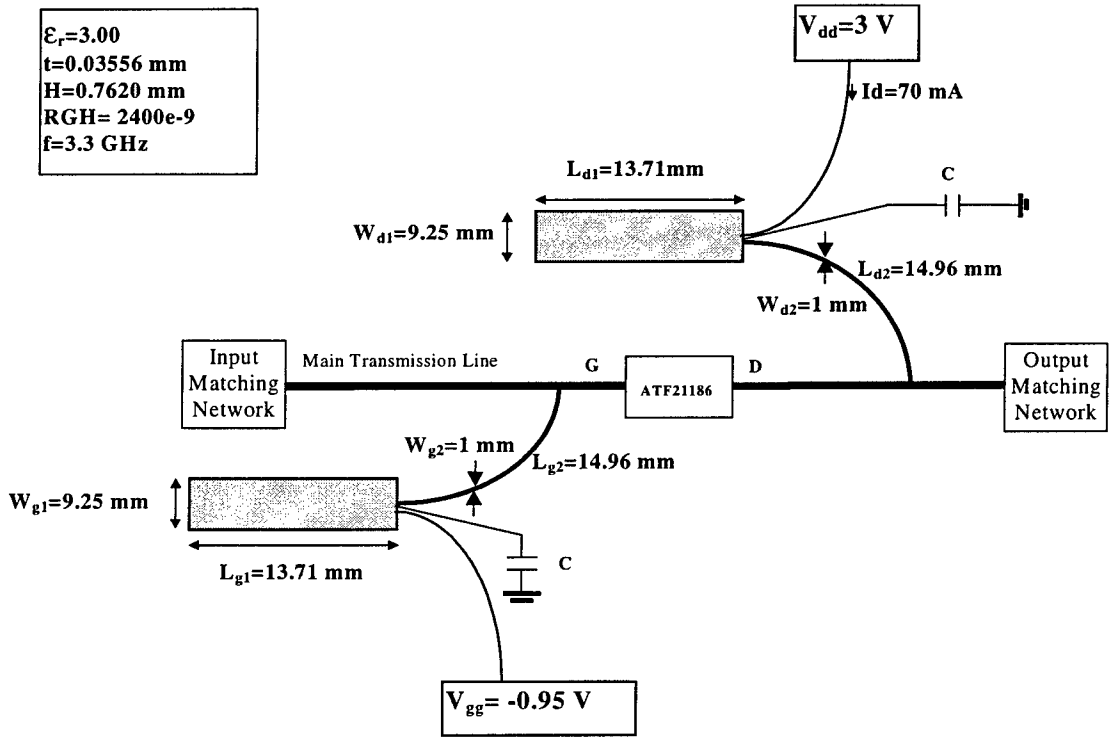


Fig.7. Configuration of the bias insertion network

V_{gs}=-0.95V ? will be adjusted *experimentally* in order to obtain a DC drain current of 70 mA under the condition V_{ds}=3V.

3.2. DC+AC Circuit Diagram

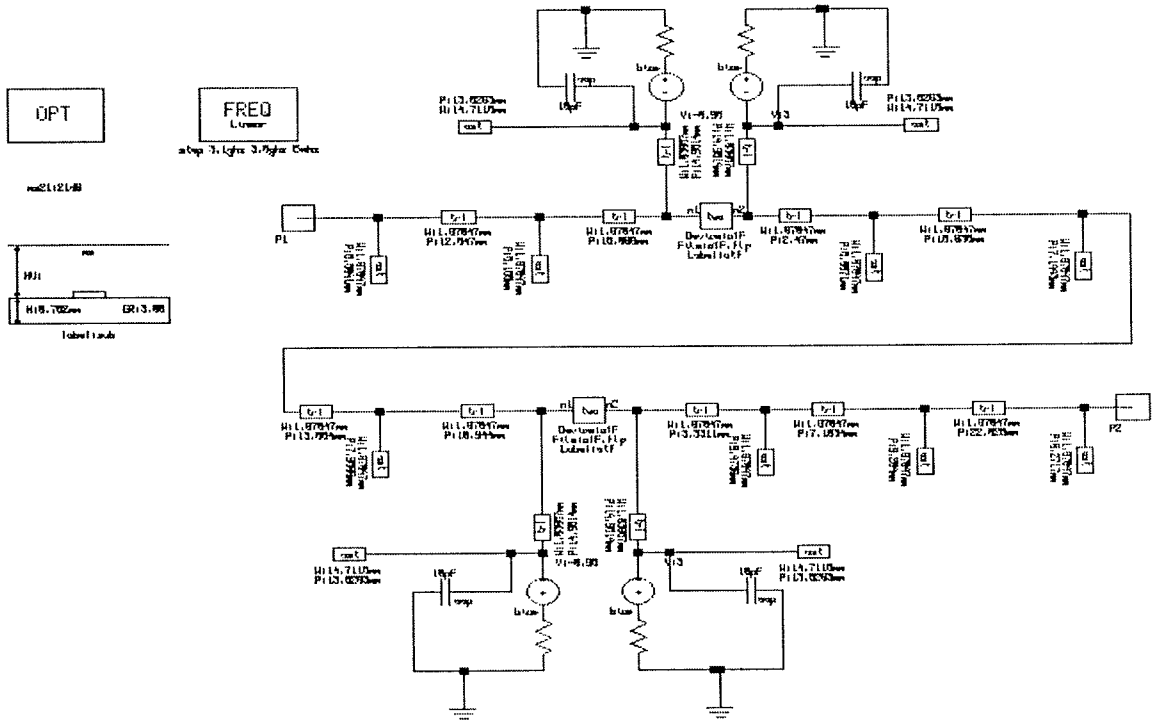


Fig.8. DC+AC Network (See table 1 and Fig.7 for physical dimensions)

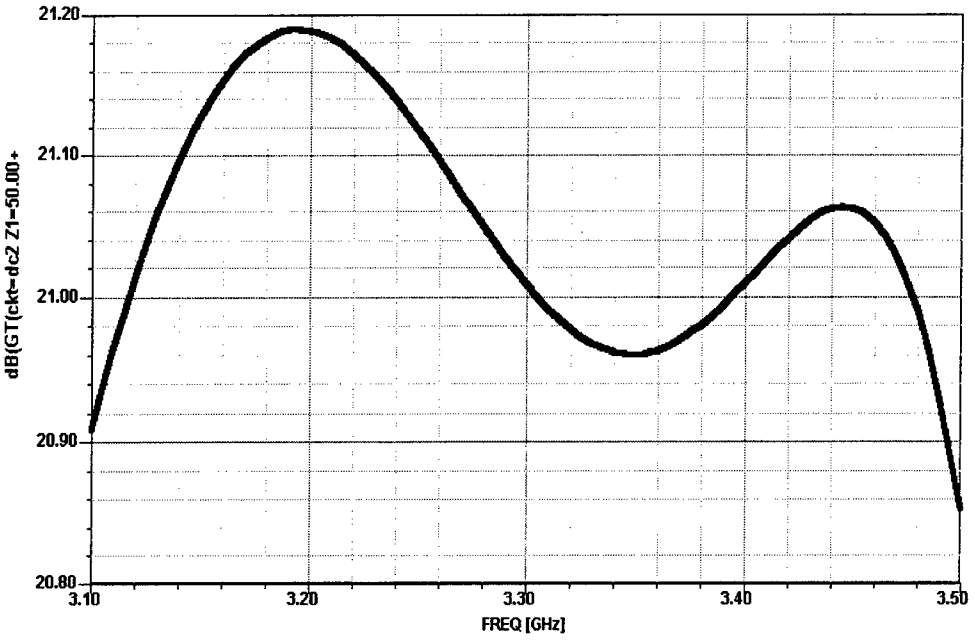


Fig.9. G_t(dB) of the two-port with dc bias network

4. STATISTICAL ANALYSIS RESULTS

4.1. Analysis Specifications

Number of trials	: 100
Distribution type	: Uniform
Moment value	: 5%
Acceptance criteria	: 20.5 dB < S21 (dB) < 21 dB
START FREQUENCY	: 3.1 GHz
END FREQUENCY	: 3.5 GHz
FREQUENCY INCREMENT	: 100 MHz

The histograms for this analysis are presented in the following figures.

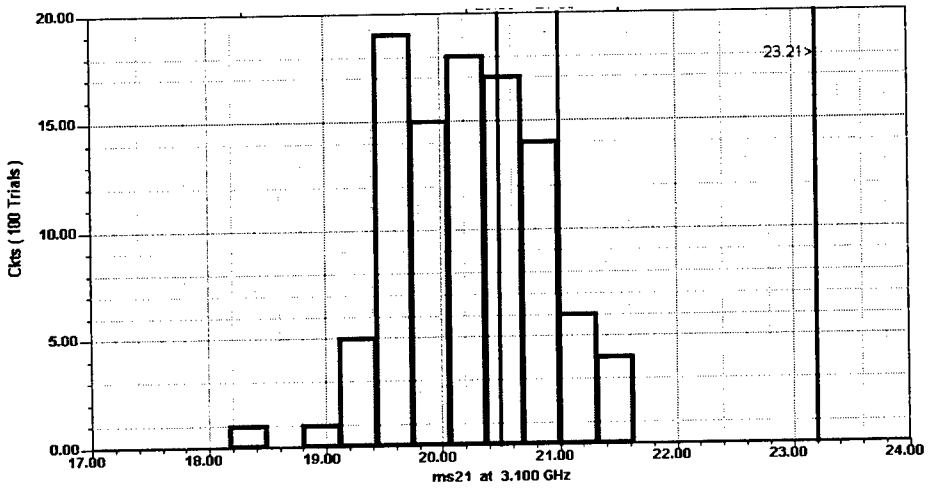


Fig. 10. Histogram for f=3.1 GHz

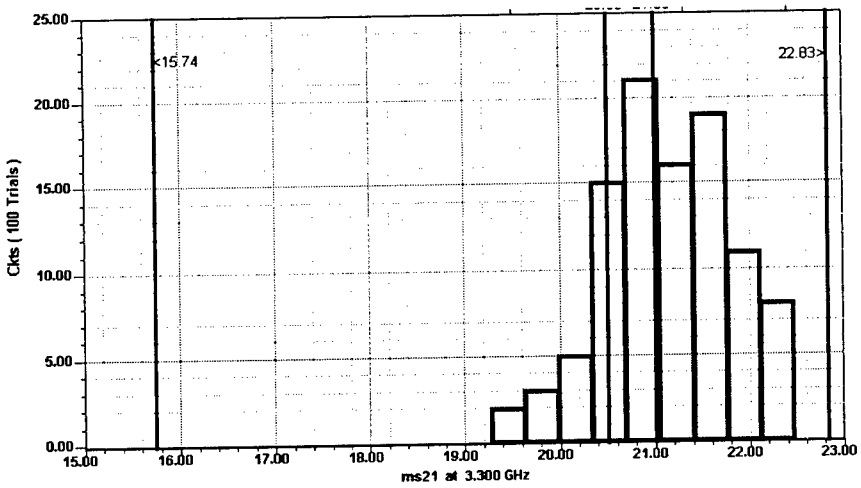


Fig. 11. Histogram for f=3.3 GHz

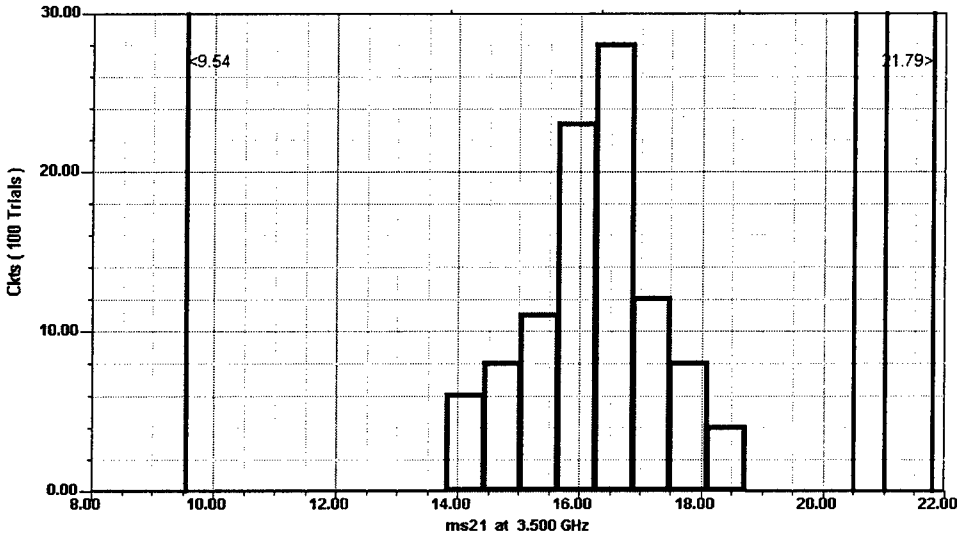


Fig. 12. Histogram for f=3.5 GHz

5. REFERENCES

[1] Microwave Transistor Amplifiers Analysis and Design, Guillermo Gonzales, 2nd Edition, 1997 by Prentice-Hall, Inc.

[2] Microwave Circuit Design Using Linear and Nonlinear Techniques, George D. Vendelin, Anthony M. Pavio, Ulrich L. Rohde, 1990 by John Wiley & Sons

[3] RF Circuit Design, Chris Bowick, 1992 by Newnes

6. ACKNOWLEDGEMENTS

I wish to thank AFOSR for summer research program and to Michael Little, Dr. Dick Michalak and Franz Haas of Rome Lab for their help with this project. Special thanks to Ertan Zencir who essentially performed all of the simulation.

INFORMATION PROTECTION TOOLS AND METHODS

Milica Barjaktarovic
Assistant Professor
Department of Electrical and Computer Engineering

Wilkes University
Stark Learning Center
Wilkes-Barre PA 18766

Final Report for:
Summer Faculty Research Program
Rome Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling AFB

and

Rome Research Site

December 1998

INFORMATION PROTECTION TOOLS AND METHODS

Milica Barjaktarovic
Assistant Professor
Department of Electrical and Computer Engineering
Wilkes University

Abstract

This report is an outcome of preparations to produce a working prototype system compliant with the Computer Security Assistance Program for the 21st Century (CSAP21) architecture. CSAP21 architecture was proposed as a way to provide integrated information protection operation for the Air Force. The architecture describes a computer system which is as automated as possible, with the task to help human personnel in daily information protection at Air Force sites. The basic premise is that software and other sensors will be placed on the network(s) of interest, and the sensor data will be used for information environment security, attack detection, attack response, and capability restoration.

The report includes an outline of several software sensors used for intrusion detection: ASIM, ISS Internet Scanner, ISS RealSecure, NetRadar and NetRanger. We also describe TIS Gauntlet firewall. For each sensor, we provide a brief description, capabilities, general structure, and data used and produced.

INFORMATION PROTECTION TOOLS AND METHODS

Milica Barjaktarovic

1. Introduction

In the current environment of distributed computing over insecure Internet, some means have to be employed to protect local computer networks and resources. Computer Security Assistance Program for the 21st Century (CSAP21)Common Intrusion Detection architecture was proposed as a way to provide integrated information protection operation for the Air Force. The architecture describes a computer system which is as automated as possible, with the task to help human personnel in daily information protection at Air Force sites. The basic premise is that software and other sensors will be placed on the network(s) of interest, and the sensor data will be used for information environment security, attack detection, attack response, and capability restoration. This report is an outcome of preparations to produce a working prototype system compliant with the CSAP21 architecture.

The report includes an outline of several software sensors used for intrusion detection: Automated Security Incident Measurement Tools (ASIM) [ASIM], ISS Internet Scanner [ISSIS], ISS RealSecure [ISSRS], NetRadar, and NetRanger [NetRan]. We also describe TIS Gauntlet firewall [Gaun]. We briefly compare and categorize each sensor. For each sensor, we provide a brief description, capabilities, general structure, and data used and produced.

2. Intrusion Defense Tools

Intrusion defense tools can be based on any combination of:

- software
- hardware (e.g. video-surveilance camera)
- human input (e.g. human analysts).

In this report we focus predominantly on the software-based tools.

We can enumerate software intrusion defense tools and mechanisms as:

- sensors (e.g. ASIM, ISS Real Secure, NetRadar, NetRanger)
- probing tools (e.g. ISS Internet Scanner, SATAN)
- firewalls (e.g. TIS Gauntlet Firewall)
- TCP wrappers
- tripwire

We will be discussing each of these tools in the following sections.

We can broadly classify software tools, based on mode of operation, as:

- sniffers (monitoring the network for passing traffic)
- log watchers (monitoring log files)
- probers (vulnerability assessment tools, attacking the system to discover vulnerabilities).

Based on physical domain of operation, tools can be classified as:

- network-based (recognizing attacks coming from/to the network)

- host-based (recognizing attacks on a specific host):
    - monitoring all system calls (secure but resource-intensive approach)
    - auditing system logs (some attacks are not recorded in system logs)

Sensors can be either designed to detect anomalies from the normal system operation; or to detect misuse of the system. Most commercial sensors detect misuse. Misuse detection tools are usually sniffers, with capability to do a combination of:

- keystroke monitoring and string matching, i.e. buzzword search (e.g. for /etc/passwd)
- attack recognition, by attack signature matching.

Examples of network based buzzword search and attack recognition tools are ASIM, NetRanger, NetRadar, and ISS Real Secure. An example of network based vulnerability assessment tool is ISS Internet Scanner. An example of host based vulnerability assessment tool is ISS Security Scanner.

2.1. Overview of Software-based Tools

ASIM: government owned, AFCRT controlled, network based sniffer. Runs in the background, sniffs external incoming/outgoing network messages for keywords predefined by AFCRT. Records messages and keystrokes sent over network, both at source and destination. Output has to be checked manually. The tool is controlled remotely by AFCRT.

ISS Set of Tools: commercial suite of software sensors by ISS. Includes the following tools: Internet Scanner, System Security Scanner, and RealSecure.

Internet Scanner: when executed, evaluates host security from the network (i.e. outside-of-host) perspective. Signature oriented; monitors traffic patters looking for chosen vulnerabilities (known weaknesses in OS). Can do probing attacks on the system.

System Security Scanner (S3): when executed, evaluates host security from an inside perspective by running on the host that is being assessed. Problems found by S3, but not by network-based scanners, include evidence of intrusion, weaknesses that allow normal users to gain administrator privileges, and weak account passwords that can be identified by password-guessing programs.

Real Secure: runs in the background, monitoring traffic and reporting real time. Signature recognition. Can respond automatically by logging, recording, or terminating actions. Can reconfigure routers and firewalls. Glitch: it saves real-time output in binary, so it is not possible to manipulate the output files (not even possible to write a script to do it, since the interface is GUI). Seems to be a more intelligent and more capable version of ASIM, oriented towards signatures instead of buzzwords.

NetRadar: Rome Research Site phase II SBIR by Tod Halberman. Network sniffer, runs in the background, monitoring traffic and reporting real time. Buzzword and signature recognition. Can take automatic action.

NetRanger: commerical network sniffer by WheelGroup Corporation. Runs in the background, monitoring traffic and reporting real time. Monitoring can be done from anywhere on the Internet. Buzzword and signature recognition. Very configurable. Can respond automatically by logging, recording, or terminating actions. Can reconfigure routers and firewalls. Based on the manuals, seems to be the most comprehensive and capable sensor. However, the performance must be evaluated to determine the effective usefulness of the tool.

Firewall: a gateway with additional capabilities. Several types of firewalls:

- packet filtering: drop packets based on source/destination IP address or port
- application-level gateway: apply special-purpose code for each desired application
- circuit-level gateway: relay TCP connection from one port to another, so that the outside network sees only the gateway and not the hosts behind it.

TIS Gauntlet Firewall: commercial application-level gateway by TIS. Includes trong authentication and encryption.

TCP wrapper: program that allows a user to log into a machine iff: he has correct userID and password, AND the source machine is on the host.allow list of the destination machine (i.e. "white list").

Tripwire: public-domain file integrity checker, a utility that compares a designated set of files against information stored in a previously generated database.

2.1.1. Definitions

Attack signature: a pattern of misuse based on one or more events. Can be atomic or composite.

Atomic signature: event based on a single ICPM packet at a specific point in time.

Composite signature: attack based on a series of events, e.g. ping sweep. Can be:

- stateless: the attack signature must be defined regardless of the order or the duration between atomic events
- stateful: attack defined based on well-defined event sequences.

(Attack) signature recognition: can be either context- or content-oriented.

Context-oriented signatures consist of known network service vulnerabilities that can be detected by inspecting packet headers. These include SATAN, TCP Hijacking, and IP spoofing attack profiles.

Content-oriented signatures require the inspection of (binary) data fields within a packet to determine if an attack or policy violation has occurred at the application level. These include e-mail and Web attack profiles.

Network Based Sniffers

Pros:	Cons:
relatively easy to install and run	false alarm rate can be high
consume relatively little resources	many attacks can be undetected
	mostly for TCP and UDP traffic only
	could slow down network if too much data is passed from the sensor

Buzzword Search Tools

Pros:	Cons:
the easiest to implement and run	high false alarm rate - buzzwords occur in normal traffic
	there might not be a physical place for the sensor to be installed to monitor the whole site
	cannot monitor encrypted messages

Attack Recognition Tools

Pros:	Cons:
higher efficiency than string matching tools	the tool must be supplied with attack patterns, and the patterns must be updated
	signature recognition must be robust enough to detect variations of the same attack pattern

Host Based Tools

Pros:	Cons:
many attacks are host-based	each host on a network must have a tool installed

Table 2-1 - Comparison of various tool types

Tool	Description	Traffic Monitored	Networks	Platform(s)	Mode of operation
ASIM vs.1.7	packet sniffer (content only); recognizes buzzwords	TCP/IP, some UDP	Ethernet, FDDI	SunSparc 5 SunOS4.1.4 Solaris2.4, 5,5.1 Linux 2.0.0	batch; some real time
ISS Internet Scanner vs.5	packet sniffer signature recognition (content and context); recognizes from-the-network attack patterns; performs fake network attacks	seems like TCP/IP and many others	Ethernet, token ring	Windows NT Solaris SunOS Linux HP-UX AIX	scan
ISS System Secure	checks individual hosts from the inside			Solaris SunOS Linux HP-UX AIX IRIX	scan
ISS Real Secure vs.2.0	packet sniffer; signature recognition (content and context); looks for traffic patterns	TCP, UDP, ICMP; Windows networking	Ethernet, Fast Ethernet, Token Ring	Windows NT Solaris Sparc5	real time; batch
NetRadar	packet sniffer; looks for buzzwords and traffic patterns	TCP, some UDP	XXX	Solaris	real time; batch
NetRanger vs.1.2.2	packet sniffer; signature recognition (content and context); looks for traffic patterns and string matches	TCP/IP, ICMP, UDP	Ethernet, Fast Ethernet, FDDI	Solaris 2.5.1	real time; batch
TIS Gauntlet Firewall	application-level gateway; strong authentication; encryption		Ethernet only for turnkey; for others any network interface supported by OS	Windows NT Solaris, IRIX, HP-UX, BSD	

Table 2-2 - Tools Summary

Note: ASIM is the only tool that cannot take automatic action.

3. Data Used for Sensor Input/Output

We will give an overview of data at the input and output of ASIM, ISS suite of tools, NetRadar, NetRanger, and TIS Gauntlet firewall.

3.1. ASIM

Capabilities: for every TCP message that traveled across the external boundary of RL from midnight to midnight, ASIM records the actual message and every keystroke related to that message. ASIM chooses a set of messages and reports on them (i.e. provides a message log for each chosen message). The 24-hr log is examined during the following day.
Implemented as Bourne-shell and C scripts.
For the purpose of brevity, we will call "message" and "message log" the entire log associated with an actual message.
Attacks recognized: SYN, probe scanning (e.g. SATAN attack).

ASIM input:

set at AFCRT:
which nets to monitor, and for which buzzwords
rules for generating real-time alerts
which hosts not to monitor
services monitored:
TCP traffic (telnet, rlogin, ftp, smtp, rsh, rexec, tcpmux, systat, netstat, sunrpc,finger, ntp, NeWS, printer, uucp, pcserver, ingreslock, whois, hostnames, uucp-patch); UDP (sunrpc, tftp, daytime, time, syslog, name); these too: game, empire, courier.
... other details

limits file:
the warning level treshold; create transcripts for all connections exceeding this level
config.file: configuration file, specifies:
buzzwords and risk number for each one

ASIM output:

connections.log file: "overall" log, all connections recorded
in.log and out.log: logs for incoming and outgoing connections
hot IP log: all hosts identified as potential or past problems
report templates

Logs are accessed via the following windows:
"main" window: selected messages from the overall log (on UNIX, stored in file /tmp/log###)
for each message in the "main" window:
separate windows for source and destination
window for all connections involving either source or destination subnets

Window format:
tabulated, user-readable text
cannot be searched or manipulated in any way through ASIM interface

Note: it is possible that there are many connections during a 24-hr period, which ASIM records in the overall log but does not include in the main window. This omission can lead to security problems.

Overall log format:

everything displayed in the windows, seems like
cannot be searched or manipulated in any way through ASIM interface

"Main Window" format:

Sorted by the service which initiated the connection:

message index (assigned message # used by ASIM for bookkeeping, to refer to individual messages; assigned sequentially as the messages come in)

message risk number (i.e. ASIM calculates how dangerous, i.e. damaging, the message can be if it is exploited). The value: 1-10, 10 being the most dangerous. ASIM calculates this value based on buzzwords found, service used, and source IP address.

DNS addresses of source and destination

Observation:

it is not possible to just click on the message display and see the message; must manually specify message index in order to see the message log.

Logs do not exist for all messages from the "main" message window. If we select to view a message log, we might have to wait for the log file to be built from the raw data.

"Source/Destination Window" format:

message index
date of message
start and end time of message
IP addresses of source and destination
DNS addresses of source and destination
service that initiated the connection
message risk number
buzzwords and their number of occurrences at source and destination
message contents at the machine which output is being shown
keystrokes at the machine which output is being shown

Note: the keystrokes should reveal user name and password.

"Other_connections_from/to_the_source/destination_subnet Window" format:

message index
message risk number
IP address of source and destination
service which initiated the connection
number of packets, incoming and outgoing
number of bytes, incoming and outgoing
duration of connection, in seconds
"*" symbol next to messages for which the log is already built
number of connections on that day

Observations:

no correlation between the source/destination windows
sometimes ASIM omits password and/or user ID in the log
sometimes the buzzwords shown in the two windows are not the same
no ability to do a keyword search through the logs
manually have to type in what message no. we want to look at
little UDP traffic, mostly TCP
seems like RL site cannot take advantage of automatic report-generating mechanism

What (should) always holds true for ASIM log messages:

source IP address belongs to a machine inside the external firewall ⇔ destination IP address belongs to a machine outside of the external firewall.

destination IP address belongs to a machine inside the external firewall ⇔ source IP address belongs to a machine outside of the external firewall.

ASIM keeps records for 2 days readily available.

3.2. ISS Internet Scanner (ISS IS)

Capabilities: when executed, evaluates host security from the network (i.e. outside-of-host) perspective. Signature oriented; monitors traffic patters looking for chosen vulnerabilities (known weaknesses in OS). Can do probing attacks on the system.

ISS IS input: IP address(es) of hosts to be scanned; IS can run ping to find hosts that are up
the policy, i.e. which tests to run
the key file, specifying scans allowed and max number of devices to be scanned.

ISS IS output: reports on the machine running the check, and email to system administrators.

Scanning:

Built-in policies:

DEC tests only
heavy scan: medium scan plus
medium scan: light scan plus most common deamons, RPC, SMTPO, FTP settings.
light scan: user accounts, password policies, auditing, basic browser security, NT registry.
web tests only
OS identification: reveal info about the scanned host, e.g. OS versions, various programs

User can define a new policy. Possible options include running all options except those that can cause interruption of service; or running only denial of service options.

Scanning options: additional output to log files, debugging info to log files, saving the original sensitive files (e.g. /etc/passwd), additional scan ports besides the default ones, max timeout value for ping, max parallel scan threads, number of parallel service scans, max ports per thread, delay between brute force checks, max number of connections during brute force checks.

Each scan is saved in the database, with time, number of hosts scanned.
ISS IS scan window provides: host IP address in text and dot format, OS type, NetBIOS name and domain, ping time.

Report:
Textual "old fashioned" reports in several flavors:

- executive:
  OS summary
  trend analysis (analysis of vulnerabilities by risk level: low, medium and high)
  vulnerability summary
  resource planning:
  host assessment: services, vulnerability, severity, corrective actions (sorted by DNS name, IP address, or OS)
  operating systems scanned: OS names, DNS names, IP addresses (sorted by DNS name, IP address, or OS)
  services found: port number, service name, service type, number of hosts running the service
  user accounts: IP address, DNS name, user ID and user name (sorted by IP address, OS, or user name)
- vulnerability assessment (sorted by DNS name, IP address, or severity)
- technician: most detailed reports on host assessment, operating systems, services, vulnerabilities
- user imported: custom format.

Reports can be sorted by scan ID, vulnerability risk level, hosts, or services.
Reports can be exported in various formats: html, MS Word, data interchange.

What (should) always holds true for ISS IS log messages:
source IP address belongs to an RL machine inside the RL's external firewall AND destination IP address belongs to a machine inside the RL's external firewall.

3.3. *ISS RealSecure*

Capabilities: runs in the background, monitoring traffic and reporting real time. Signature recognition. Responds to unauthorized or suspicious activity automatically by logging, recording, or terminating actions. Enforces Java and Active/X security policy. Enables dynamic reconfiguration of routers and firewalls based on predefined policy.

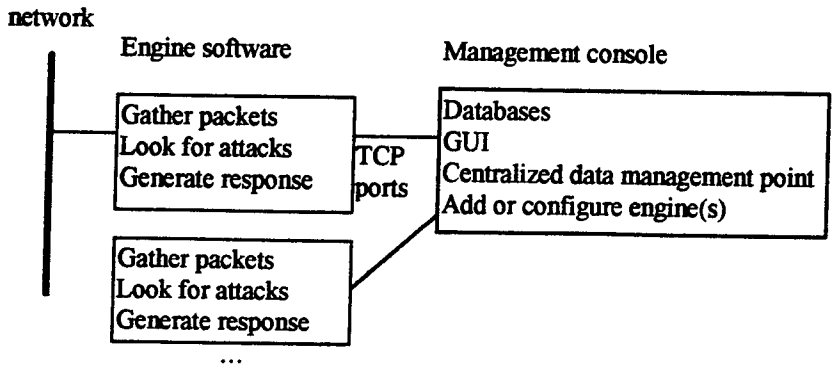


Figure 3-1 - ISS Real Secure Block Diagram: one console can control many engines

Engine configuration:

attack signatures(i.e. what network patterns are recognized)
user defined connection events
filters
user-specified actions
monitored networks
e-mail notification
SNMP traps
firewall configuration parameters (reconfigure firewall to block messages from an IP for a specified amount of time)

Console configuration:

custom reports
enable DNS lookup
strong authentication

Console display window:

title
activity tree: all current events, stored by the active engines; sorted by source host, destination host, event name; click on event to get event inspection window
high, medium, low priority windows: real time information received by the windows
session playback: keystrokes from previous sessions
reports

Event inspection window:

event name
date, time
IP addresses or DNS addresses of source and destination
IP address of engine
network protocol used
ports at source and destination
any additional info available (e.g. in POP-password event, username and subject)
actions resulting from the event

Engine window: (for every engine with which the console is communicating)

status
current policy
network traffic (packets/sec)

Types of attacks recognized:

DOS: SYN flood, oing flood, WinNuke
Unauthorized access: FTP root, email WIZ
Recons: SATAN, port scan, IP half scan
Suspicious activity: duplicate IP address, IP unknown

Options overview:

log session to database (for playback)
log raw (binary) data in a file
send email notification
kill connection (by sending TCP RST)
view session in real time on management console
lock firewall (i.e. drop packets from the attacker's address)
send SNTP trap
user predefined action, i.e. up to four actions from the following list:

Report:

Text or graphic. Sorted by events, based on event's:

priority
source IP address
destination IP address
name

Also includes: time and date, information about ? (not specified in the manual, but shown in a graph).
Text reports can include port numbers.
Custom reports are available.

3.4. NetRadar

Capabilities: records all messages and keystrokes sent over network in TCP traffic. Mainly is used for "buzzword" search. Can recognize certain attack patterns based on the type/size of message. Has some decision making ability and ability to take action.

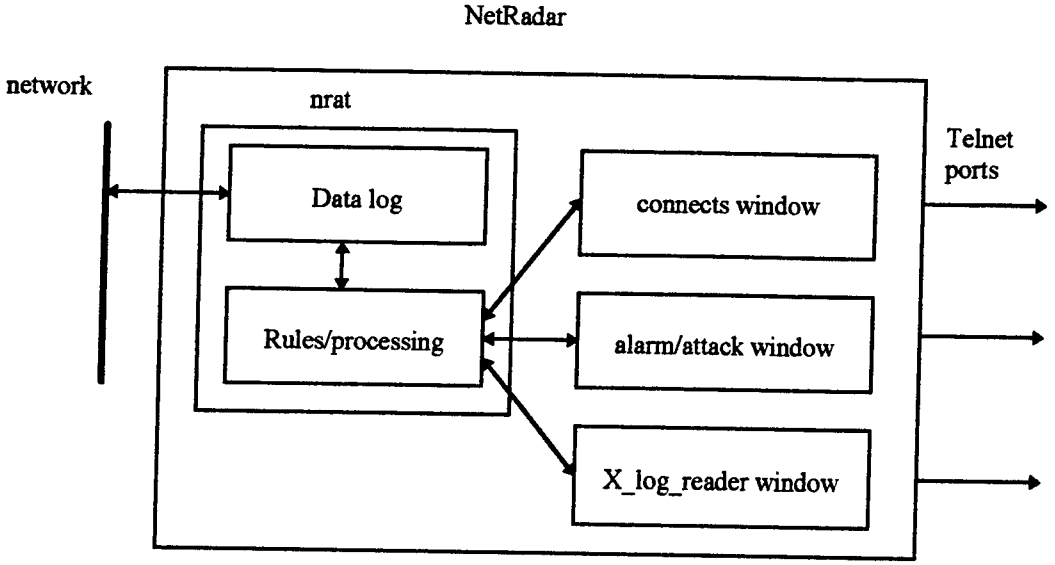


Figure 3-2 - NetRadar Architecture: each window has a log file associated with it, which can be displayed by accessing a particular telnet port

Format: nrat, alarm/attack window, connects window, x_log_reader window, stats window.

nrat:
Actual sensor, i.e. network monitor plus. Collects data, processes it, displays it, and makes some decisions about it.
nrat has: database in which it stores the data logs, some rules for processing and displaying the logs.

connects window:
Connections window; shows current connections in real time. Format:

session number
service name (e.g. TCPP_START, TCP_FULL_START, CLOSE, RESET, STOP)
IP addresses of source and destination
port number used by client to connect to the server
confidence rating that the assessment which machine is client and which machine is server is assigned correctly (0<conf<4)

This window may have lines which show:

X term size
term type
user name
user password

STR_MATCH: string matched
session number
dir: direction from which the buzzword came (CLIENT or SERVER)
index number: what security alert it was (there is a list of numbers assigned)
cont: number of occurrences of the buzzword found

Question: why this information is not displayed in alarms window by default?

Note: since NetRadar distinguishes between TCP_START and TCP_FULL_START connection, it recognizes TCP connections which were started but not opened (TCP goes through a three-way handshake with SYN and ACK packets; client attempts to open a connection by sending SYN packet, server accepts connection by sending ACK packet).

x_log_reader window:
nrat builds this window from the raw data. Format:

- time and date
- IP addresses of source and destination
- port numbers used at source and destination
- types of packets exchanged (e.g. SYN, ACK, FIN)
- login information: userID, password
- string matches and direction: buzzword(s) found and which direction they came from

Stats window:
Approximatelly every 5secs display statistics on protocol streams. Format:

- number of connections attempted
- number of connections suceeded
- two more that need to be looked up

Possible protocol streams: Rlogin, Telnet, FTP, Login, Rshell, string, IpAddrObj, TCP.

Configuration files and their content:

- ports: what ports to save information from
- patterns: buzzwords to look for (currently 78 words); if found, information about them is displayed in connections window
- string_matches: can be programmed to display the found buzzword in the attack window, if the buzzword for found present a predetermined number of times.
- session_address: what hosts to store information from
- session_port: what sessions to store from the hosts running on the ports specified in ports file

alarm/attack window:
Window with information related to possible attacks. Format:

ZONE-CHANGE: indicating change in amount of network traffic, either increasing or decreasing. Based on number of ICMP and port connections, assign a discrete number indicating the "zone". Zone could be 0,1,2, ...

- addr: IP address of sender
- statistic: ?
- old: old zone number
- new: new zone number

ATTACK: indicating attack

- attacker: IP address of attacker
- target: IP address of target
- type: number assigned to this type of attack (range unknown)
- subtype: number assigned to this subtype of attack (range unknown)

Attacks recognized: teardrop, SYNflood, Sunkill, ping of death, land attack and latierra, smurf, pepsi, and syndrop.

Observations:
NetRadar reports on a greater number of connections than ASIM.
NetRadar cannot be used to detect attack patterns which do not require more than one attack packet, because it examines one message at a time for buzzwords, size, or some other message parameters. For example, if an attack were to constitute of sending "ls" then "telnet", NetRadar would not be able to detect it. It would store the messages, though, so another decision-making tool, such as Gensym, could examine the data and find the pattern.

3.5. NetRanger

Capability: real-time monitoring from a host located anywhere on the Internet. Recognizes attacks based on signature and string matching.

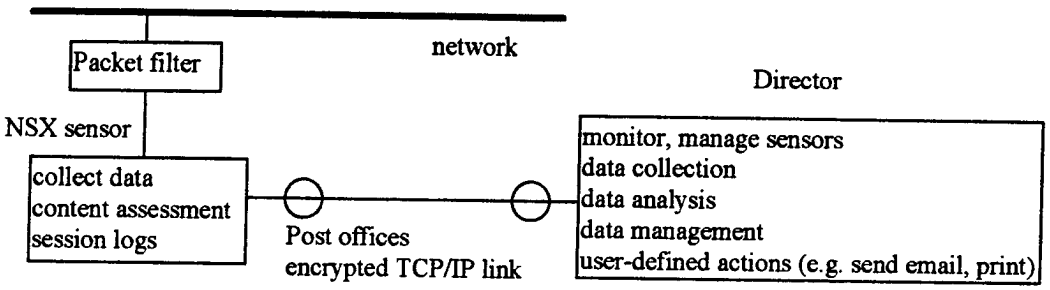


Figure 3-3 - NetRanger Block Diagram

Caveats:

sensor must be installed behind a router
must be installed with its own router filters, replacing the existing router filters.
TCP connection logs are in binary.

Data:
Security information presented graphically:

where the sensors are
color-coded security level of each sensor.

Security levels are: normal, marginal, critical.
Security levels are assigned based on the risk level of events at that host, where each event (i.e. attack pattern) is assigned a (user-configurable) risk level.
Data analysis: for patterns and trends, based on time, space and event.

Attacks recognized:
Each attack is assigned risk level called "alarm value", which is an integer 1-5, 5 being the maximum.
Context based attacks: source routing, ICPM sweeps, fragmented ICMP traffic, large ICMP traffic, TCP port sweep, SYN flood, TCP hijacking, UDP port scan, SATAN scan, ...
Content based attacks: smail, sendmail invalid recipient, sendmail invalid sender, sendmail recon, TFTP password, DNS HINFO request, DNS zone transfer, DNS request for all records, RPC port registration and unregistration, RPC dump, proxied RPC request, NFS mound request, rexd request, YP attacks, loadmodule attacks, WWW phf attack, WWW cgi-bin attack, "+ +"in a TELNET session, any matched string specified by the user, ...

Alarm set: multiple alarms are received that are identical in all respects but in timestamp and sequence number.

Interesting features:

Use HP OpenView for information display.
Use third-party tools for report writing and multi-dimension analysis.
Console features:
view event list
find DNS address
reject IP traffic from a specified source for a specified time
save information to a file (e.g. to send it via email)
email notification
reconfiguring the router to shun packets.

NetRanger Logs:

Can be configured to be stored locally on NSX or remotely at the Director.
Stored in flat files, in two flavors:

- event logs: any event or a piece of event generated by a user or a deamon (i.e. commands, keystrokes, errors). ASCII file. Includes logging of events, commands, and errors.
- IP session logs: whole TCP connections or parts of them, recorded in entirety or partially, based on triggering events (e.g. IP address, string match) during a specified time. Binary file.

Event Log format:

data type
record ID
date and time, local and GMT
application, host and organization ID of source (integer numbers)
direction of traffic (IN, OUT)
risk level of event (integer greater than 0)
event signature and subsignature (assigned integer numbers)
protocol name
IP address of source, destination, and router
ports at source and destination
string matched

Error Log Format:

data type
record ID
date and time, local and GMT
application, host and organization ID of source (integer numbers)
error message (text string)

Command Log Format:

data type
record ID
date and time, local and GMT
application, host and organization ID of source (integer numbers)

application, host and organization ID of requestor (integer numbers)
command executed (text string)

IP Session Log Format: (in raw binary):
timestamp
packet length
IP packet

3.6. TIS Gauntlet Firewall

Capabilities: Application-level gateway. Strong authentication, encryption. Does not permit any connectionless protocol like UDP or ICMP across the firewall. Checks for viruses. Real-time alerts. Source code is available.

Features:
TELNET, Rlogin, FTP, HTTP, Gopher, SMTP, NNTP, RSH proxies;
X11 gateway, authentication server, Java and ActiveX blocking, URL screening, SSL, SHTTP, POP3, others

Logs:
Human-readable UNIX syslogs. *(what for NT?)*
Logging (by default):
All operating system kernel warnings and errors
All file system warnings and errors
All attempted accesses to network services; whether successful, supported, rejected source routed addresses and ICMP redirects.
All successful network accesses:
source and destination address
service
time and day
number of bytes transferred
commands accessed (FTP)
URLs accessed (HTTP)
All interactions with the user authentication server subsystem

Reports:
Summary report: use of each service summarized by user and usage.
top 20 users
how many times they connected
how many bytes transferred
Exception report: what events should be excluded from the reports

Users should write scripts to look through the logs.
Manual says: "more extensive logging, etc will be available through third party products in mid-1997." In Aug 1998 - seems like the page was not updated.

4. Summary of Sensor Inputs/Outputs/Behavior

After examining several sensors, we can see certain similarities between them and deduce what features could be desirable.
All sensors that we examined base their analysis on keystrokes and/or attack patterns executed. The sensors monitor TCP traffic and some UDP traffic (e.g. ping).

Most sensors display (usually as GUI):

- IP and DNS addresses of source and target
- user name, password if possible (use nslookup, whois)
- date, time
- protocol used
- ports used
- number of bytes transferred, length of connection
- strings/buzzwords were recognized
- attack patterns were recognized

Most sensors can:

- monitor certain ports/hosts/IPs
- remember "bad" IPs
- store at least parts of connections monitored
- search/manipulate stored data
- provide real time alerts
- provide email notification

Services desired include:

- discard most obvious false alarms
- block desired IPs (usually by dynamically reconfiguring firewalls)
- terminate connection with desired IPs
- see all messages involving a desired IP address
- assign a meaningful risk number to each message
- click on event displayed to get more detailed information about it
- provide detailed instructions on how to fix the problem
- predictions on how the attack and fixes can impact the system
- draw (color-coded) map of network and sensors.

References

[ASIM] Trident Data Systems. "ASIM User Guide," vs1.7, June 1997.

[EPIC] EPIC Web Server, http://ares.iwt.rl.af.mil.

[Gaun] "TIS Internet Firewall Kit," TIS, http://www.tis.com/prodserv.

[ISSIS] "ISS Internet Scanner manual." ISS Download Center, http://download.iss.net/eval.

[ISSRS] "ISS Real Secure Manual." ISS Download Center, http://download.iss.net/eval.

[NetRan] WhhelGroup Corporation, "Net Ranger," http://wheelgroup.com/netrangr/1netrang.html.

OUTLIER RESISTANT DS-SS SIGNAL PROCESSING

Stella N. Batalama
Assistant Professor
Department of Electrical and Computer Engineering
E-mail: batalama@eng.buffalo.edu

State University of New York at Buffalo
201 Bell Hall
Buffalo, NY 14260

Final Report for:
Summer Faculty Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Laboratory

September 1998

ADAPTIVE ROBUST SPREAD-SPECTRUM RECEIVERS

Stella N. Batalama
Assistant Professor
Department of Electrical and Computer Engineering
State University of New York at Buffalo
E-mail: batalama@eng.buffalo.edu

Abstract

We consider the problem of adaptive robust detection of a Spread-Spectrum (SS) signal in the presence of unknown correlated SS interference and additive non-Gaussian noise. The proposed general SS receiver structure is comprised by a vector of adaptive chip-based non-linearities followed by an adaptive Auxiliary-Vector linear tap-weight filter. The novel characteristics of our approach are as follows: First, the non-linear receiver front-end adapts itself to the unknown prevailing noise environment providing robust performance over a wide range of underlying noise distributions. Second, the adaptive Auxiliary-Vector linear tap-weight filter that follows the non-linearly processed chip-samples, allows rapid adaptation for SS interference suppression with limited data record. We examine a clipper-type, as well as a puncher type and a Hampel-type non-linearity. Numerical and simulation results demonstrate the performance of the proposed approach and offer comparisons with the conventional Minimum-Variance-Distortionless-Response (MVDR) filter, as well as the MVDR filter preceded by a vector of adaptive chip-based non-linearities.

OUTLIER RESISTANT DS-SS SIGNAL PROCESSING

Stella N. Batalama

I. Introduction

A receiver is considered outlier resistant if its performance does not deteriorate severely when small deviations from the model assumptions occur. There is a vast amount of research performed in the past on the subject of outlier resistant statistical methods with straightforward applications to engineering design [1].

Theoretically, a statistical algorithm (receiver) is designed to optimize a criterion for a given data model. Regardless of the assumed model the principle of outlier resistant design makes it necessary to evaluate the performance of a system not only at the model distribution but in a neighborhood of the model. This is equivalent to the more realistic assumption that a model is known approximately rather than accurately. Such reasoning gives rise to the ϵ-contamination class where the assumed model lies within a stated distance of a known distribution. Other physical interpretations that support the validity of the ϵ-contamination model talk about outlying events that often corrupt a nominal statistical platform. Thus, it seems imperative to try to devise receivers that are not strongly affected by outliers. For a given data model there is a great variability in the outlier resistance capabilities among different optimization criteria and receiver designs. To add on the complications involved when outlier resistant performance is sought, it remains to consider a non-asymptotic working framework where conclusions are less clear-cut due to the higher random variability of the pertinent design parameters.

Signal detection in the presence of impulsive channel noise has been considered in [2]-[9], while direct-sequence spread-spectrum (DS-SS) signal detection under similar channel characteristics has been studied in [11]-[13]. In [11]-[10] receiver proposals involve the use of either a conventional matched filter or a majority-vote receiver (that is a hard-limiter non-linearity per chip followed by a matched-filter operation matched to the signature of the user of interest). In [10] it is reported that neither one of the above proposals is universally effective against the combination of SS interference and non-Gaussian noise. In [13] adaptive receivers are proposed that are comprised by a vector of adaptive chip-based non-linearities followed by an adaptive linear tap-weight filter. The structures proposed in [13] tap the relative merits of both non-linear and linear signal processing and exhibit superior BER performance in the presence of both impulsive and SS interference. In particular, the non-linear receiver front-end adapts itself to the unknown prevailing noise environment providing robust performance over a wide range of underlying noise distributions, while the adaptive linear tap-weight filter that follows the non-linearly processed chip-samples combats effectively the SS interference. This report extends the work in [13] in the following aspects. It shifts the receiver figure of merit to superior bit error rate (BER) performance in short-data-record realistic situations, still, however, in the presence of SS interference and impulsive noise. This is in contrast to the developments in [13] where an asymptotic framework is considered and the focus is rather on the merits of combining adaptive non-linear chip-based pre-processing with adaptive filtering. The new receiver is designed based on a maximum cross-correlation principle that exhibits favorable characteristics in short data support situations. In addition, the new system model accounts for multipath (in [13] no multipath is considered). With respect to the latter consideration, the adaptive non-linear pre-processors are generalized to comply and fully utilize the new system characteristics while

the linear filter post-processor exhibits some form of RAKE diversity thus taking advantage of the multipathed reception. Finally an additional non-linear pre-processor of Hampel-type is considered that combines the characteristics of the non-linearities studied in [13].

The rest of the report is organized as follows. In Section II we present the system model. In Section III we derive the receiver non-linearities pertinent to the new model and develop the linear maximum cross-correlation (auxiliary-vector) -type filter that follows them. Algorithmic implementation issues as well as numerical results and simulations are presented and discussed in Section IV.

II. System Model

Without loss of generality, and for simplicity of presentation only, we consider a conventional DS-SS set-up and signal model. Our system involves a signal of interest with code $\mathbf{S}_0$ of length L chips (the symbol period is T, the chip period is T_c, and $L = T/T_c$), M multipaths, (K-1) multipathed DS-SS interferers with signatures $\mathbf{S}_k$, $k = 1, \ldots, K-1$, and non-Gaussian (impulsive) interference. For notational simplicity, we choose a synchronous set-up and avoid the incorporation of neighboring cell interference. We assume that the multipath spread is of the order of a few chip intervals and since the signal is bandlimited to $B = 1/T_c$, the tap-delay line channel model has taps spaced at chip intervals T_c. The chip-synchronous received signal vector during a symbol interval that results after conventional chip-matched filtering and sampling at the chip rate, comprises of $L + M - 1$ samples and is given by

$$\mathbf{r} = \sum_{k=0}^{K-1} b_k \sum_{m=1}^{M} c_{k,m} \sqrt{E_k} \mathbf{S}_{k,m} + \mathbf{n}, \tag{1}$$

where, with respect to the k-th SS signal, b_k and E_k are the transmitted bit and energy, respectively, $\{c_{k,m}\}$ are the coefficients of the frequency-selective slowly fading channel modeled

as independent zero-mean complex Gaussian random variables that remain constant within a bit interval. $\mathbf{S}_{k,m}$ is the signature of the k-th SS signal due to its m-th path (we assume 0-padding for the expansion of $\mathbf{S}_k$ to $\mathbf{S}_{k,m}$) and $\mathbf{n}$ is assumed to be complex non-Gaussian impulsive noise.

An equivalent representation of the received signal $\mathbf{r}$ is given by

$$\mathbf{r} = \sqrt{E_0} b_0 \mathbf{w}^{(0)}_{\text{R-MF}} + \mathbf{I} + \mathbf{n}, \tag{2}$$

where $\mathbf{w}^{(0)}_{\text{R-MF}} = \sum_{m=1}^{M} c_{0,m} \mathbf{S}_{0,m}$ is the effective signature of the SS signal of interest (signal-0), and $\{c_{0,m}\}$ are the channel coefficients. In (2), $\mathbf{I}$ identifies comprehensively both the Inter-Symbol and the SS interference. We use the subscript R-MF in the effective signature notation to make an association with the RAKE Matched-Filter receiver that is known to correlate the signature $\mathbf{S}_0$ with M size-L shifted windows of the received signal (that correspond to the M paths of the channel) appropriately weighted by the conjugated channel coefficients, $c_{0,m}, m = 1, \ldots, M$ [14].

III. Receiver Design Considerations

1. Distribution-free non-linear pre-processors.

In this section we generalize the distribution-free non-linearities considered in [13] for the multipath model adopted. We also propose to study an additional Hampel-type non-linearity. The non-linearities used in this work are defined as follows:

Clipper-type non-linearity

$$g_1(x) \triangleq \begin{cases} x, & \text{if } |x| \leq c, c > 0 \\ c\frac{x}{|x|}, & \text{if } |x| > c, c > 0. \end{cases} \tag{3}$$

Puncher-type non-linearity

$$g_2(x) \triangleq \begin{cases} x, & \text{if } |x| \leq c, c > 0 \\ 0, & \text{if } |x| > c, c > 0. \end{cases} \tag{4}$$

Hampel-type non-linearity

$$g_3(x) \triangleq \begin{cases} x, & \text{if } |x| < a, a > 0 \\ a\frac{x}{|x|}, & \text{if } a \leq |x| < b, a, b > 0 \\ \frac{c-|x|}{c-b} a\frac{x}{|x|}, & \text{if } b \leq |x| \leq c. a, b, c > 0 \\ 0, & \text{otherwise .} \end{cases} \tag{5}$$

In all the above non-linearities, x is a complex number and $|x|$ denotes the norm of x. Also, the linear region of the non-linearities has the effect of passing the observations undistorted. The non-linear regions either completely reject (remove) or "correct" the observation. The latter is considered as an adjustment of the norm while maintaining the phase. The parameters $a, b,$ and c involved in the above definitions are (positive) cutoff parameters to be determined adaptively.

2. MS-type linear post-processing

A solution for the suppression of SS-interference, intersymbol interference and non-Gaussian (impulsive) noise that assumes the least information about these signal disturbances (i.e. only the effective code of the signal of interest is assumed known) can be obtained by an adaptive linear filter designed according to a Minimum Variance Distortionless Response (MVDR) criterion preceded by a vector of adaptive chip-based non-linearities. This MVDR-type receiver is a generalization of the receiver in [13] for the multipath case. The linear post-processor is a RAKE-type filter that minimizes the output variance subject to the constraint that the filter

remains distortionless in the normalized "effective" direction/signature $\mathbf{w}^{(0)}_{||\text{R-MF}||}$ of the user of interest. It is given by

$$\mathbf{w}^{(0)}_{\text{g,R-MVDR}} \triangleq \frac{\mathbf{R_g}^{-1}\mathbf{w}^{(0)}_{||\text{R-MF}||}}{\mathbf{w}^{(0)H}_{||\text{R-MF}||}\mathbf{R_g}^{-1}\mathbf{w}^{(0)}_{||\text{R-MF}||}}, \tag{6}$$

where $\mathbf{R_g}$ denotes the autocorrelation matrix of the size $(L+M-1)$ non-linearly processed received signal that includes all M resolvable paths (L is the signature length), i.e. $\mathbf{R_g} = E\{\mathbf{g(r)g}^H\mathbf{(r)}\}$. To form, however, the MS-optimum adaptive filter $\mathbf{w}^{(0)}_{\text{g,R-MVDR}}$, exact knowledge of the size $(L+M-1)$ data autocorrelation matrix $\mathbf{R_g}$ is required. Since $\mathbf{R_g}$ is unknown we may use available non-linearly processed received data to form a sample-average estimate $\hat{\mathbf{R}}_\mathbf{g} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{g}(\mathbf{r}_n)\mathbf{g}^H(\mathbf{r}_n)$ and then proceed with Sample-Matrix-Inversion (SMI). In this case

$$\hat{\mathbf{w}}^{(0)}_{\text{g,R-MVDR}} = \frac{\hat{\mathbf{R}}_\mathbf{g}^{-1}\mathbf{w}^{(0)}_{||\text{R-MF}||}}{\mathbf{w}^{(0)H}_{||\text{R-MF}||}\hat{\mathbf{R}}_\mathbf{g}^{-1}\mathbf{w}^{(0)}_{||\text{R-MF}||}}. \tag{7}$$

In the above expressions, the normalized effective signature is given by

$$\mathbf{w}^{(0)}_{||\text{R-MF}||} = \frac{\mathbf{w}^{(0)}_{\text{R-MF}}}{\left\|\mathbf{w}^{(0)}_{\text{R-MF}}\right\|}. \tag{8}$$

Finally, the decisions are made according to

$$\hat{b}_0 = sgn(Re[\hat{\mathbf{w}}^{(0)H}_{\text{g,R-MVDR}}\mathbf{g(r)}]), \tag{9}$$

where sgn$(\cdot)$ denotes the sign operation.

3. Proposed algorithmic developments

It is known that batch SMI adaptive procedures outperform recursive constraint-LMS adaptive implementations of the MVDR solution in terms of small-sample convergence characteristics [15]. Apart from the computational complexity of the matrix inversion operation, $N \geq L+M-1$ non-linearly processed data samples are required for $\hat{\mathbf{R}}_\mathbf{g}$ to be invertible with probability 1 [16] and in fact data sizes many times the filter length $(L+M-1)$ are necessary for $\hat{\mathbf{w}}^{(0)}_{\text{g,R-MVDR}}$ to

approach reasonably well the performance characteristics of the "ideal" $\mathbf{w}^{(0)}_{g,\text{R-MVDR}}$ filter, where "ideal" refers to the use of an asymptotically large data record. Unfortunately, for typical L and M values and data transmission rates, filter adaptation over that many symbol intervals may not keep up with typical channel fluctuation (fading) rates [15]. This discussion brings us to the core concept of this work. That is to develop a SS-receiver that (i) operates in presence of both SS and impulsive interference, (ii) taps the merits of both non-linear signal processing and adaptive filtering, (iii) exhibits inherently low computational complexity and, at the same time, (iv) maintains superior short data record filtering performance.

To start gaining some insight with respect to the proposed algorithmic framework, we consider a canonical representation of the linear filter post-processor as depicted in Fig. 1. In this figure we look at the class of scalar parameterized FIR linear post-processing filters of the form

$$\mathbf{w}^{(0)}_{\text{PROP}} \triangleq \mathbf{w}^{(0)}_{||\text{R-MF}||} - \mu_{\text{g}}\mathbf{G}_{\text{g}}, \tag{10}$$

where μ_{g} is an arbitrary complex-valued parameter, and $\mathbf{G}_{\text{g}}$ is an arbitrary "auxiliary" complex vector of dimension $L+M-1$ that is orthonormal with respect to $\mathbf{w}^{(0)}_{||\text{R-MF}||}$:

$$\mathbf{G}^{H}_{\text{g}}\mathbf{w}^{(0)}_{||\text{R-MF}||} = 0 \quad \text{and} \quad ||\mathbf{G}_{\text{g}}|| = 1. \tag{11}$$

Both μ_{g} and $\mathbf{G}_{\text{g}}$ are to be designed. Bit detection is made as follows:

$$\hat{b}_0 = sgn(Re[\hat{\mathbf{w}}^{(0)H}_{\text{PROP}}\mathbf{g}(\mathbf{r})]). \tag{12}$$

We note that both $\mathbf{G}_{\text{g}}$ and μ_{g} are parametrized by the $L \times 1$ vector, $\mathbf{g}(\cdot)$, of non-linearities. At this point, it is important to note that for any $\mathbf{G}_{\text{g}}$ satisfying the above constraints, the filter $\mathbf{w}^{(0)}_{||\text{R-MF}||} - \mu_{\text{g}}\mathbf{G}_{\text{g}}$ is already "distortionless" in the $\mathbf{w}^{(0)}_{||\text{R-MF}||}$ direction of interest since $\mathbf{w}^{(0)H}_{\text{PROP}}\mathbf{w}^{(0)}_{||\text{R-MF}||} = 1$. For an arbitrary but fixed auxiliary vector $\mathbf{G}_{\text{g}}$ that satisfies the above constraints, the filter in (10) can be viewed as scalar parameterized and optimized with respect to the complex scalar

μ_g only. It is very easy to find the MS-optimum value of μ_g: It is the value that minimizes the MSE $E\{|\mathbf{w}_{||\text{R-MF}||}^{(0)H}\mathbf{g}(\mathbf{r}) - \mu_g^* \mathbf{G}_g^H \mathbf{g}(\mathbf{r})|^2\}$ between the points (a) and (c) in the block diagram filter representation in Fig. 1.

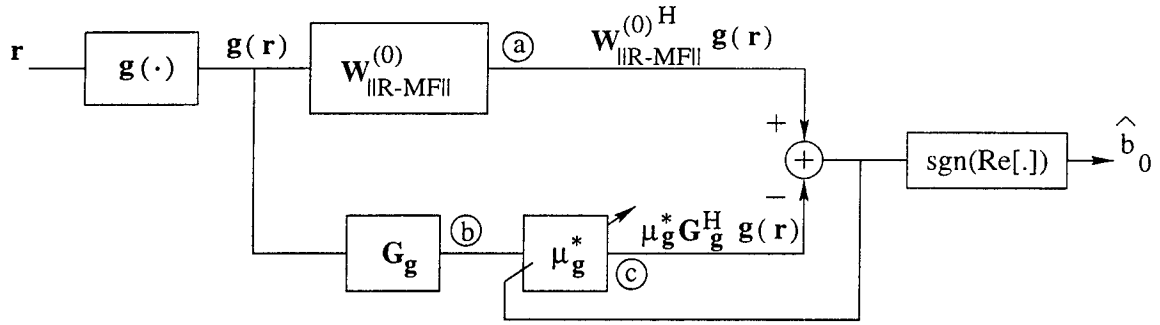


Fig. 1: General structure of the proposed receiver.

The solution to the first minimum output variance problem can be obtained by direct differentiation of the variance expression. The solution to the second MSE problem can be obtained by direct application of the optimum linear MS estimation theorem [17]. It is simply

$$\mu_g = \frac{\mathbf{G}_g{}^H \mathbf{R}_g \mathbf{w}_{||\text{R-MF}||}^{(0)}}{\mathbf{G}_g{}^H \mathbf{R}_g \mathbf{G}_g}, \tag{13}$$

where $\mathbf{R}_g = E\{\mathbf{g}(\mathbf{r})\mathbf{g}(\mathbf{r})^H\}$.

Now let us return to the problem of selecting an auxiliary vector subject to an appropriately chosen criterion. Of course, there exists a vector $\mathbf{G}_g$ in the subspace orthogonal to $\mathbf{w}_{||\text{R-MF}||}^{(0)}$ for which $\mathbf{w}_{\text{PROP}}^{(0)}$ and the ideal/asymptotic $\mathbf{w}_{g,\text{R-MVDR}}^{(0)}$ filter coincide. This solution, however, is not desirable since no bias/variance trade takes place and we are back at "Square 1": The ideal/asymptotic MVDR solution. The selection criterion for the auxiliary vector $\mathbf{G}_g$ that we propose to study in this research work is motivated by the MSE interpretation of the filter in Fig. 1. It is the maximization of the magnitude of the cross-correlation between the points (a) and (b) in Fig. 1 subject to the constraints in (11):

$$\mathbf{G}_g = \arg\max_{\mathbf{G}} |E\{\mathbf{w}_{||\text{R-MF}||}^{(0)H}\mathbf{g}(\mathbf{r})(\mathbf{G}^H\mathbf{g}(\mathbf{r}))^*\}| = \arg\max_{\mathbf{G}} |\mathbf{w}_{||\text{R-MF}||}^{(0)H}\mathbf{R}_g\mathbf{G}|, \tag{14}$$

$$\text{subject to } \mathbf{G}_g^H \mathbf{w}_{||\text{R-MF}||}^{(0)} = 0 \text{ and } \mathbf{G}_g^H \mathbf{G}_g = 1 .$$

The magnitude cross-correlation criterion function $|\mathbf{w}_{||\text{R-MF}||}^{(0)\,H} \mathbf{R}_g \mathbf{G}_g|$ and the orthonormality constraints are all phase invariant. So, to avoid unnecessary ambiguities and without loss of generality, we can identify the unique auxiliary vector that is a solution to our constraint optimization problem and make $\mathbf{w}_{||\text{R-MF}||}^{(0)H} \mathbf{R}_g \mathbf{G}_g$ non-negative real ($\mathbf{w}_{||\text{R-MF}||}^{(0)H} \mathbf{R}_g \mathbf{G}_g \geq 0$). This is the vector

$$\mathbf{G}_g = \frac{\mathbf{R}_g \mathbf{w}_{||\text{R-MF}||}^{(0)} - (\mathbf{w}_{||\text{R-MF}||}^{(0)H} \mathbf{R}_g \mathbf{w}_{||\text{R-MF}||}^{(0)}) \mathbf{w}_{||\text{R-MF}||}^{(0)}}{\| \mathbf{R}_g \mathbf{w}_{||\text{R-MF}||}^{(0)} - (\mathbf{w}_{||\text{R-MF}||}^{(0)H} \mathbf{R}_g \mathbf{w}_{||\text{R-MF}||}^{(0)}) \mathbf{w}_{||\text{R-MF}||}^{(0)} \|} . \tag{15}$$

We want to comment at this point that the proposed maximum cross-correlation criterion (MCC) corresponds to the maximization of the numerator of the second term in the variance expression at the output of the $\mathbf{w}_{\text{PROP}}^{(0)}$ filter,

$$E\{|\mathbf{w}_{\text{PROP}}^{(0)H} \mathbf{g}(\mathbf{r})|^2\} = \mathbf{w}_{||\text{R-MF}||}^{(0)H} \mathbf{R}_g \mathbf{w}_{||\text{R-MF}||}^{(0)} - \frac{|\mathbf{G}_g^H \mathbf{R}_g \mathbf{w}_{||\text{R-MF}||}^{(0)}|^2}{\mathbf{G}_g^H \mathbf{R}_g \mathbf{G}_g}, \tag{16}$$

subject to the constraints in (11). Maximization of the second ratio as a whole under $\mathbf{G}_g{}^H \mathbf{w}_{||\text{R-MF}||}^{(0)} = 0$ and unconstrained norm, takes us again back to "Square 1": The ideal/asymptotic MVDR solution $\mathbf{w}_{g,\text{R-MVDR}}^{(0)}$. In addition, the fact that no matrix inversion is explicitly required or implicitly assumed, is expected to result in superior performance in realistic small data record adaptive implementations.

Finally, we propose to generalize the MCC approach for processing with multiple auxiliary vectors. In particular, we propose to consider conditional optimization, i.e. to determine the new vector, say $\mathbf{G}_2$, and the new scalar, say μ_2, by assuming that the main direction branch $\mathbf{w}_{||\text{R-MF}||}^{(0)}$ (Fig. 1) has been updated and fixed to $\mathbf{w}_{||\text{R-MF}||}^{(0)} - \mu_1 \mathbf{G}_1$, where μ_1 and $\mathbf{G}_1$ were determined from the previous step. This way the optimization at each step will be of the same very low complexity and will exhibit all the desired properties identified so far.

The proposed algorithmic developments are summarized in the following Proposition.

Proposition 1 *Given the nonlinearity* $\mathbf{g}(\cdot)$ *then:*

(i) The maximum cross-correlation linear tap-weight RAKE-type filter $\mathbf{w}_{\text{PROP}}^{(0)}$ *is given by*

$$\mathbf{w}_{\text{PROP}}^{(0)} \triangleq \mathbf{w}_{||\text{R-MF}||}^{(0)} - \mu_{\text{PROP}}\mathbf{G}_{\text{g}}, \tag{17}$$

where

$$\mu_{\text{PROP}} = \frac{\mathbf{G}_{\text{g}}^{H}\mathbf{R}_{\text{g}}\mathbf{w}_{||\text{R-MF}||}^{(0)}}{\mathbf{G}_{\text{g}}^{H}\mathbf{R}_{\text{g}}\mathbf{G}_{\text{g}}}, \tag{18}$$

and

$$\mathbf{G}_{\text{g}} = \frac{\mathbf{R}_{\text{g}}\mathbf{w}_{||\text{R-MF}||}^{(0)} - (\mathbf{w}_{||\text{R-MF}||}^{(0)H}\mathbf{R}_{\text{g}}\mathbf{w}_{||\text{R-MF}||}^{(0)})\mathbf{w}_{||\text{R-MF}||}^{(0)}}{\|\mathbf{R}_{\text{g}}\mathbf{w}_{||\text{R-MF}||}^{(0)} - (\mathbf{w}_{||\text{R-MF}||}^{(0)H}\mathbf{R}_{\text{g}}\mathbf{w}_{||\text{R-MF}||}^{(0)})\mathbf{w}_{||\text{R-MF}||}^{(0)}\|}. \tag{19}$$

(ii) The generalized multiple auxiliary vector RAKE-type linear post-processor is given by

$$\mathbf{w}_{\text{PROP-Mul}} = \mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{P}\mu_i\mathbf{G}_i, \tag{20}$$

where both the scalar parameters $\{\mu_i\}$ *as well as the vectors* $\{\mathbf{G}_i\}$ *can be determined recursively by the following expressions:*

$$\mu_{p+1} = \frac{\mathbf{G}_{p+1}^{H}\mathbf{R}_{\text{g}}(\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{p}\mu_i\mathbf{G}_i)}{\mathbf{G}_{p+1}^{H}\mathbf{R}_{\text{g}}\mathbf{G}_{p+1}}. \tag{21}$$

and

$$\mathbf{G}_{p+1} = \frac{\mathbf{R}_{\text{g}}(\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{p}\mu_i\mathbf{G}_i) - [\mathbf{w}_{||\text{R-MF}||}^{(0)H}\mathbf{R}_{\text{g}}(\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{p}\mu_i\mathbf{G}_i)]\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{j=1}^{p}[\mathbf{G}_j^{H}\mathbf{R}_{\text{g}}(\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{p}\mu_i\mathbf{G}_i)]\mathbf{G}_j}{\|\mathbf{R}_{\text{g}}(\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{p}\mu_i\mathbf{G}_i) - [\mathbf{w}_{||\text{R-MF}||}^{(0)H}\mathbf{R}_{\text{g}}(\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{p}\mu_i\mathbf{G}_i)]\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{j=1}^{p}[\mathbf{G}_j^{H}\mathbf{R}_{\text{g}}(\mathbf{w}_{||\text{R-MF}||}^{(0)} - \sum_{i=1}^{p}\mu_i\mathbf{G}_i)]\mathbf{G}_j\|}. \tag{22}$$

For both (i) and (ii) the covariance matrix $\mathbf{R}_{\text{g}}$ *is given by* $\mathbf{R}_{\text{g}} = E\{\mathbf{g}(\mathbf{r})\mathbf{g}(\mathbf{r})^{H}\}$. □

Idenitfying a stopping rule based on a tractable performance metric that will terminate the auxiliary vector generation at any number below the full space dimension is a research subject of particular importance and one of our future research directions.

IV. Simulation Studies

The optimization of the filter parameters follow the general principles of the MSE-BER optimization introduced in [13]. In particular, let the receiver output be denoted by $u(\mathbf{r}, c)$, i.e. $u(\mathbf{r}, c) \triangleq sgn(Re[\mathbf{w}_c^H \mathbf{g}_c(\mathbf{r})])$ where c is the cutoff parameter of the non-linearity involved (eg. puncher or clipper type). Then, the distortion measure $\rho(\mathbf{r}_0, \mathbf{r}_1; c)$ defined at the receiver output by

$$\rho(\mathbf{r}_0, \mathbf{r}_1; c) \triangleq \frac{1}{2}\{\pi_0[1 + u(\mathbf{r}_0, c)] + \pi_1[1 - u(\mathbf{r}_1, c)]\}\,, \tag{23}$$

where $\pi_0 = \pi_1 = 1/2$ are the a priori probabilities of hypothesis H_0 ($b_0 = -1$) and H_1 ($b_0 = +1$), exhibits an average value equal to the BER, i.e.

$$P_e(c) = E\{\rho(\mathbf{r}_0, \mathbf{r}_1; c)\}\,, \tag{24}$$

where $\mathbf{r}_0$ and $\mathbf{r}_1$ correspond to received data vectors from H_0 and H_1 respectively. Exploiting the property in (24), we can develop a stochastic gradient technique that adjusts the cutoff parameter c to minimize the probability of error at the output of the receiver, as follows:

$$c_{n+1} = c_n - \alpha_n x_n(c_n)\,, \tag{25}$$

where

$$x_n(c) \triangleq \frac{1}{2d_n}[\rho(\mathbf{r}_{0,n}, \mathbf{r}_{1,n}; c + d_n) - \rho(\mathbf{r}_{0,n}, \mathbf{r}_{1,n}; c - d_n)]\,, \tag{26}$$

and $\{d_n\}$ and $\{\alpha_n\}$ are monotonically decreasing sequences of positive numbers such that $\sum \alpha_n = \infty$, $\sum \alpha_n d_n < \infty$, and $\sum \alpha_n^2 d_n^{-2} < \infty$. A similar gradient alogrithm can be obtained when more than one cutoff parameters are involved in the non-linearity used (eg. Hampel-type).

We examine a DS-SS transmission with 5 SS interfering signals and impulsive noise. The normalized cross-correlation of the interfering signals with the signal of interest is approximately 25% and the energies are equal to 9, 10, 11, 12, 13dB. The system processing gain is 63.

Fig.2 emphasizes the BER performance of the proposed receiver (implemented with one and two auxiliary vectors) and the MVDR receiver as a function of the norm of the projection of the SS signal of interest into the interfering subspace, when the clipper-type or puncher-type non-linear processing front-end is employed. Receiver needed ensemble averages are replaced by sample-average estimates based on a data-record of 150 samples. The channel noise is modeled according to the the outlier model, described in the following paragraph with the difference that no multipath is considered for this figure (ie. all functions are real).

The multipath fading channel is modeled as having 4 paths with zero mean complex Gaussian fading coefficients of variance 0.5 (i.e. $E\{|c_{k,m}|^2\} = 0.5$) for all paths and SS signals. The impulsive channel noise is modeled according to the ϵ-mixture disturbance model given by the following expression:

$$f_\epsilon(x) = (1-\epsilon)f_0(x) + \epsilon f_1(x)\,, \tag{27}$$

where $\epsilon \in [0,1]$ accounts for the probability under which the noise is $f_1(\cdot)$ distributed. The nominal pdf $f_0(\cdot)$, is taken to be complex Gaussian with variance $\sigma_0^2 = 1$. The "contaminating" pdf, $f_1(\cdot)$, is assumed to be either a complex Gaussian pdf with variance $\sigma_1^2 = \gamma^2\sigma_0^2$ ($\gamma^2 = 1000$) or a delta function at $\pm\infty \pm j\infty$. In the former case the resulting model is called Gaussian ϵ-mixture while in the latter case outlier model.

We compare the BER performance of the proposed receiver (implemented with one and two auxiliary vectors) with that of the MVDR receiver when the clipper-type, Hampel-type or puncher-type non-linear processing front-end is employed. The BER of the conventional MVDR is also included as a reference point. The results are averages over 100 independent channels.

In Fig. 3 we present the BER performance of all receivers as a function of the energy of

the SS signal of interest for the Gaussian ϵ-mixture model. Fig. 4 carries out the same studies for the outlier model. For both figures, receiver needed ensemble averages are replaced by sample-average estimates based on a data-record of 150 samples.

Figs. 5 and 6 plot the BER as a function of the size of the data record for the same set-up as in Figs. 3 and 4. Fig. 5 shows the BER performance for the Gaussian ϵ-mixture model, while Fig. 6 deals with the outlier model.

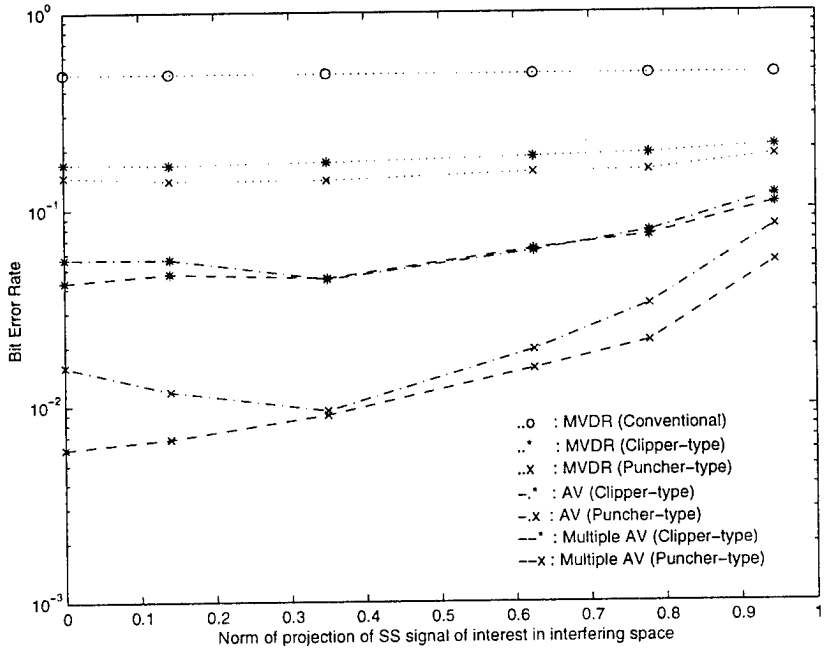


Fig. 2: Bit error rate as a function of the norm of the projection of the SS signal of interest into the interfering subspace in the presence of outlier noise.

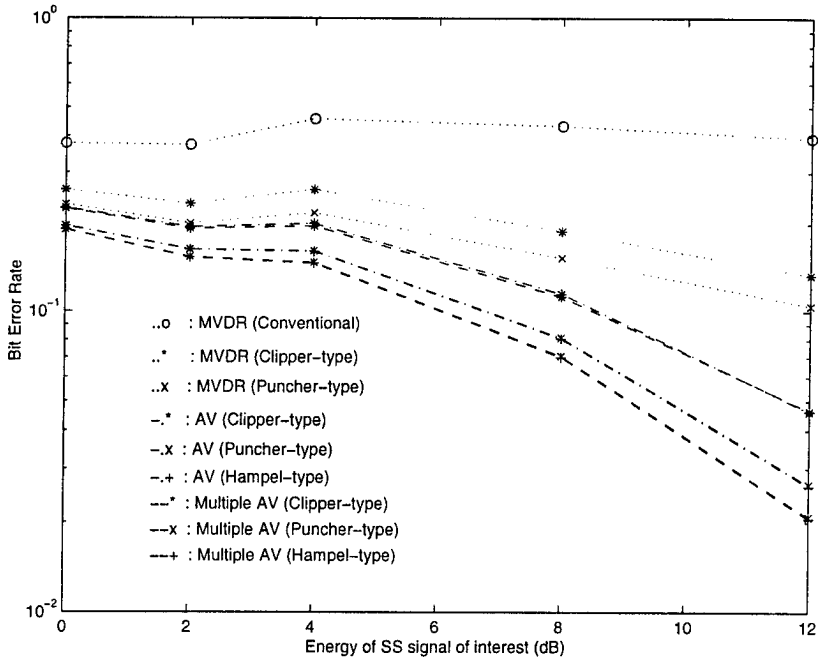


Fig. 3: Bit error rate as a function of the energy of the SS signal of interest in the presence of Gaussian ϵ-mixture noise.

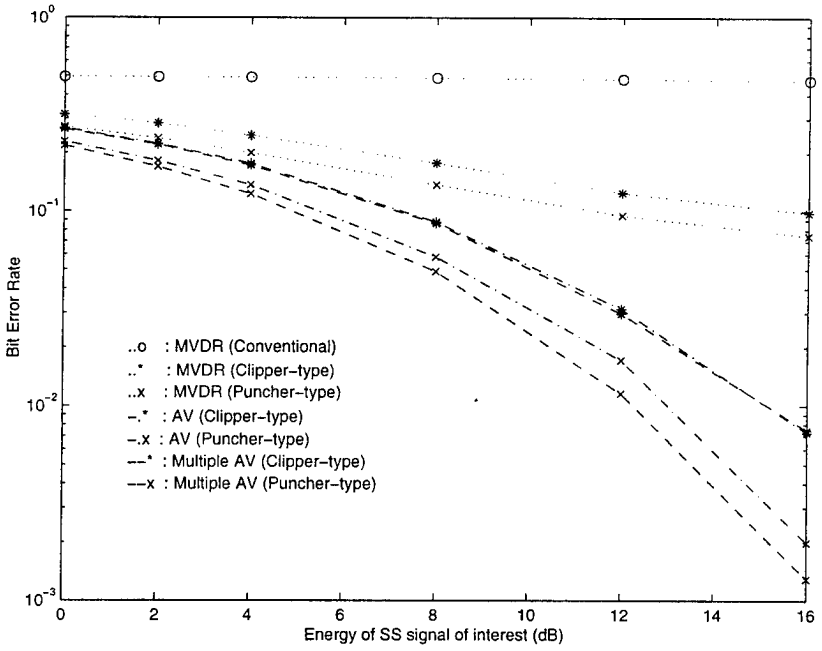


Fig. 4: Bit error rate as a function of the energy of the SS signal of interest in the presence of outlier noise.

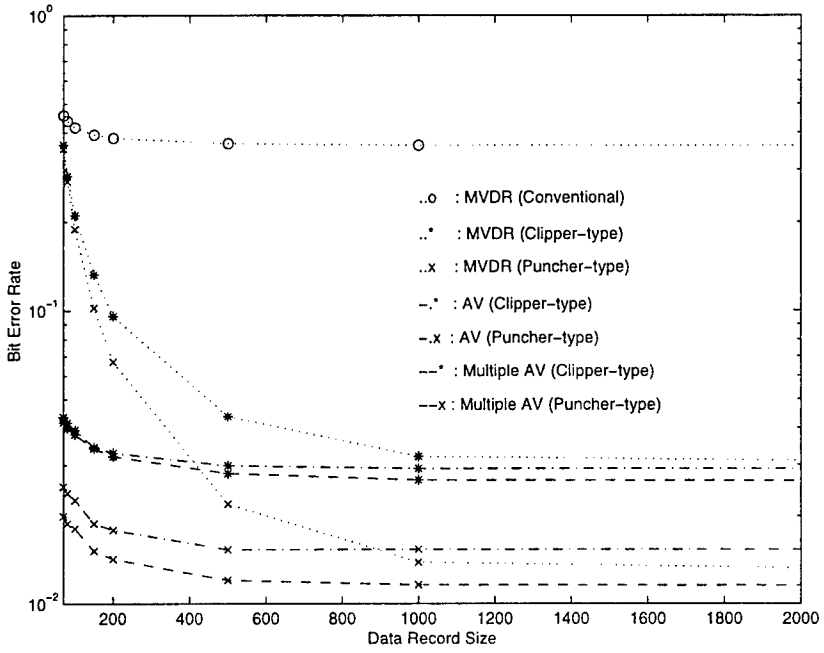


Fig. 5: Bit error rate as a function of the training set size in the presence of Gaussian ϵ-mixture noise.

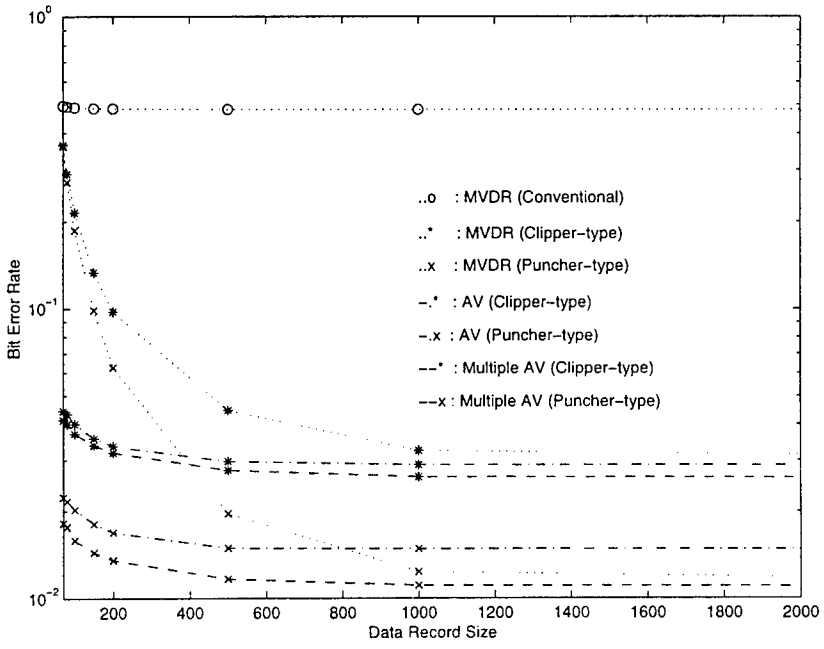


Fig. 6: Bit error rate as a function of the training set size in the presence of outlier noise.

References

[1] Peter J. Huber, *"Robust Statistics,"* Wiley, 1981.

[2] A. Tsihrintzis and C. L. Nikias, "Performance of optimum and suboptimum receivers in the presence of impulsive noise modeled as an α-stable process", *IEEE Trans. Commun.*, vol., No., pp. , Apr. 1995.

[3] A. D. Spaulding and D. Middleton, "Optimum reception in an impulsive interference environment-Part I: Coherent detection", *IEEE Trans. Commun.*, vol. COM-25, No. 9, pp. 910-923, Sept. 1977.

[4] J. H. Miller and J. B. Thomas, "Detectors for discrete-time signals in non-Gaussian noise", *IEEE Trans. Info. Theory* , vol. IT-18, No. 2, pp. 241-250, Mar. 1972.

[5] A. D. Spaulding and D. Middleton, "Optimum reception in an impulsive interference environment-Part II: Incoherent reception", *IEEE Trans. Commun.*, vol. COM-25, No. 9, pp. 924-934, Sept. 1977.

[6] S. A. Kassam, "Locally robust array detectors for random signals", *IEEE Trans. Info. Theory*, vol. 24, No. 3, pp. 309-316, 1978.

[7] S. Blum, "Locally optimum distributed detection of correlated random signals based on ranks", *IEEE Trans. Inform. Theory*, IT-42, pp. 931-942, May 1996.

[8] A. D. Spaulding, "Locally optimum and suboptimum detector performance in a non-Gaussian interference environment", *IEEE Trans. Commun.*, vol. COM-33, No. 6, pp. 509-517, June 1985.

[9] A. B. Martinez,P. F. Swaszek, and J. B. Thomas, "Locally optimal detection in multivariate non-Gaussian noise", *IEEE Trans. Info. Theory*, vol. IT-30, No. 6, pp. 815-822, Nov. 1985.

[10] B. Aazhang and H. V. Poor, "Performance of DS/SSMA communications in impulsive channels-Part II: Hard-limiting correlation receivers", *IEEE Trans. Commun.*, vol. 36, 1988

[11] B. Aazhang and H. V. Poor, "An analysis of nonlinear direct-sequence correlators", *IEEE Trans. Commun.*, vol. 37, No. 7, pp. 723-731, July 1989.

[12] B. Aazhang and H. V. Poor, "Performance of DS/SSMA communications in impulsive channels-Part I: Linear correlation receivers," *IEEE Trans. Commun.*, vol. COM-35, No. 11, pp. 1179-1188, Nov. 1987.

[13] S. N. Batalama, M. Medley and I. N. Psaromiligkos "Adaptive Robust Spread-Spectrum Receivers," *IEEE Trans. Commun.*, accepted for publication.

[14] J. G. Proakis, *Digital Communications*, New York: Mcgraw Hill, 1989.

[15] L. C. Godara, "Applications of antenna arrays to mobile communications, Part II: Beamforming and direction-of-arrival considerations," *IEEE Proceedings*, vol. 85, pp. 1195-1245, Aug. 1997.

[16] E. J. Kelly, "An adaptive detection algorithm," *IEEE Trans. Aerospace and Electronic Syst.*, vol. 22, No. 1, pp. 115-127, Mar. 1986.

[17] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms*. Englewood Cliffs, NJ: Prentice Hall, 1995.

MODELING AND SIMULATION OF MEMS RESONATORS

Digendra K. Das
Associate Professor
Department of Mechanical Engineering Technology

SUNY Institute of Technology at Utica/Rome
P.O. Box 3050
Utica NY 13504-3050

Final Report for:
Summer Faculty Research Program
Air Force Research Laboratory
Rome NY

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base
Washington DC
and
Air Force Research Laboratory
Rome NY

August 1998

MODELING AND SIMULATION OF MEMS RESONATORS

Digendra K. Das
Associate Professor
Department of Mechanical Engineering Technology
SUNY Institute of Technology of Utica/Rome

Abstract

The Micro-electro-mechanical Systems (MEMS) CAD Software MEMCAD 4.0 and CFD-ACE+ were reviewed for applications in the development of MEMS models. The software MEMCAD 4.0 was used to develop two MEMS Resonator models. Results from the simulation models were used to study the behavior of the Macromodel using the software SIMULINK and MATLAB.

MODELING AND SIMULATION OF MEMS RESONATORS

Digendra K. Das

INTRODUCTION

In recent years, a significant amount of papers have been published on the design and development of Micro-electro-mechanical Systems (MEMS). Several types of these devices are now produced commercially by many industries. As a consequence of this, there is a growing need for the development of CAD tools for MEMS. To address this need of the industry, the Microcosm Technologies, Inc. has developed a CAD tool package and their latest package has been trade named as MEMCAD 4.0. A similar CAD package, trade named CFD-ACE+, has been developed by the CFD Research Corporation for applications in the design and development of.Microfluidic systems.

The objective of the 1998 Summer Faculty Research Program project was to review these two CAD tool packages and develop a modeling and simulation technique to design a couple of MEMS Resonators. In the following sections a brief review of these software and the development of the MEMS Resonator models are described.

MEMCAD

Microcosm Technologies, Inc.'s MEMCAD design automation software suite provides device design and manufacturing analysis solutions for Micro-electro-mechanical Systems (MEMS). MEMCAD is a commercial product developed from a Massachusetts Institute of Technology (MIT) research project on software techniques for development of Micro-electro-mechanical Systems. The first version of this software (MEMCAD 3.0) was released in early 1996. Since then the software package has been significantly enhanced and in the current SFRP, the latest version (MEMCAD 4.0) of the package has been used.

Functionally, MEMCAD is divided into four major components (Ref. 1)

1) Device Creation
2) Device Modeling
3) Managed Simulation, and
4) Visualizer

1) Device Creation

The package allows the user to create MEMS designs in a variety of ways. An integrated 2D layout editor supports basic drawing and viewing functions for simple designs. As an alternative, designs can be created with an independent layout tool and imported into MEMCAD 4.0 in either CIF or GDSII format.

Once the mask layers have been defined, the MEMCAD 4.0 process editor allows users to create a flow, simulating the foundry process that will fabricate the MEMS design. Materials, mask dimensions, and etch profiles are entered into the process flow using a simplified sequence of deposit and etch steps. Materials declared in the process flow can be entered into a materiel property database (MPD), which can store all the parameters needed to fully characterize the materials of choice.

The completed mask/process description is converted into a new file format for import into MemBuilder, a shell with enhanced script functions for the SDRC's (Structural Dynamics Research Corporation) I-DEAS (Integrated Design Engineering Analysis Software) Master Series modeling and meshing toolset. The imported files are automatically rendered into a 3D model in I-DEAS, where it can be further modified if desired. As an alternative, the mask/process steps can be bypassed, and all drawing, dimensioning, material assignment, and rendering can be completed within I-DEAS, or imported directly into I-DEAS from another source with a compatible file format.

The I-DEAS tools are then used to create a mechanical mesh for the finite element solvers used in MEMCAD 4.0. The completed design is output in a universal file format for use by the MEMCAD 4.0 solvers.

2) Device Modeling

The heart of MEMCAD is the solvers, an extensible set that includes electrostatic, mechanical, thermal, fluidic, and coupled solution components. A desired solver or group of solvers is chosen and a series of windows allow users to set up file paths, solver configuration parameters, and boundary conditions.

The MEMCAD solvers are rich in features and allow many different types of solutions to be computed. Some examples are:

- Deformation from applied pressure or forces,
- Capacitance and charge calculations,
- Capacitance calculations with models having dielectric layers,
- Coupled electromechanical solutions,
- Thermomechanical solutions

Completed solutions are available in table matrix format or can be viewed using the MEMCAD 4.0 visualizer. The Visualizer color maps the solution onto the rendered 3D model, allowing users to view localized areas of stress, thermal gradients through a solid model or deformations due to applied boundary conditions.

A list of MEMCAD 4.0 solvers are given below:

1. MemCap: The electrostatic solver
2. MemMech: The Mechanical solver
3. Co-Solve-EM: The coupled electromechanical solver
4. MemTherm: The Thermal solver
5. MemCFD: The fluidic solver
6. Mem Package: The package effect analyzer
7. MemE Therm: The electrothermal specialty solver
8. MemPZR: The peizo resistive specialty solver
9. NetFlow: The electrokinetic fluidic specialty solver

3) Managed Simulation

Once a solver analysis of a MEMS device has been completed, parameters may be iterated to create a solution set for graphing or additional analysis. The Simulation Manager allows user to create batch run iterations by defining one or more parameters to be varied during the simulation. Thus, a MEMS model can be run through a sequence of solver iterations using the simulation manager to vary model dimensions, material parameters, temperature or voltage values, or a variety of other boundary conditions without altering the base model created during the Device Creation phase of a MEMCAD design. The competed solution set can be graphed or

viewed in the Visualizer as an animated sequence of events to clearly define trends and variations.

The Managed Simulation also includes the following Modules:

a) Auto Spring, and

b) AutoMM

a) Auto Spring: The Spring Constant Extractor.

The Auto Spring module allows the extraction of multi-dimensional, non-linear spring behavior from complex MEMS tether designs in a simple and automatic way. These spring behaviors are converted to a set of fitting parameters that can then be applied to any MEMS device surface as a mathematical boundary condition. This technique can considerably simplify the model building process and produce a numerically efficient model for simulation.

b) Auto MM: The Behavioral Model Generator.

The AutoMM module automatically generates lumped macromodels for MEMS sub-components from 3D MEMCAD 4.0 simulation results. It will automatically run the simulations required to characterize the mechanical and electrostatic lumped behavior. It will then fit these behaviors to polynomials and generate a net list for system level simulation.

4) Visualizer

The solver output for most MEMS models includes a significant amount of data stored in a MEMCAD 4.0 native mbif file format for analysis and post-processing. The solvers include tables of primary result values, along with graphing capability where appropriate. But the visualizer tool enables users to map these results graphically on the original model and visually analyze the solution.

The Visualizer enables viewing of several different types of stresses, Thermal Variations and temperature gradients, pressures, current density, and many other parameters in addition to electrostatic fields and mechanical deformations. Selected parameters are color-mapped onto the model in 3D, allowing a complete surface analysis of the model after the solution is complete. An adjustable slice plane and selected probing of interior bricks allow the entire volume to be visualized.

The Resonator Model

The following two MEMS Resonator models were developed during the 1998 SFRP:

1) Single Beam Resonator: Aluminum only
2) Composite Beam Resonator: Aluminum and Oxide

All parameter extractions and simulations were performed using MEMCAD 4.0, the MEMS design and simulation software package described in the previous section. The modeling and simulation methodology adopted is described briefly:

First, the physical model of the resonator was created by using the SDRC's software I-DEAS, which is an integral part of the MEM Builder module of MEMCAD 4.0. Next, a finite element mesh was generated and mechanical and electrostatic boundary conditions applied.

The coupled electrostatic/mechanical simulation of the model was done by using the AutoMM module of MEMCAD 4.0. The following steps were involved in this simulation process (Ref 2):

Extraction of Electrostatic Spring Data

The sub-module Auto Spring was used for the extraction of electrostatic spring data. It varies the position and angular orientation of the movable parts of the electro-mechanical system over the desired ranges of operation along the desired degrees-of-freedom. (We used only two degrees-of-freedom-Y and Z translations for the models developed). For each of these states, the MEMCAD Boundary-element (BEM) based electrostatic solver, MemCap, was used to determine the capacitance matrix which related the system's conductors. The simulation data from each MemCap run (i.e., each position state) were collected in a file and fitted with a multi-degree polynomial equation (up to fourth order). The polynomial co-efficients were stored for use during Macromodel generation.

Extraction of Mechanical Spring Data

The Auto Spring module was also used to extract the mechanical spring data in a similar manner over the desired ranges of operation along the desired degrees-of-freedom. The MEMCAD finite-element (FEM) based mechanical solver, MemMech, was used to determine

the reaction forces on the tether ends. The results of all MemMech runs were collected in a file and fitted to polynomial equation as above.

Determination of Mass and Controid

The MEMCAD module MemMass was used to compute the mass of the movable mechanical structures and their controids. These data were stored in a file, for use during the Macromodel generation.

Generation of the Macromodel

The data generated in the previous sub-module simulations were used to form the following macromodel equation of motion:

$$M\ddot{x} = Kx + B\dot{x} + \frac{v^2}{2}\frac{dc}{dx}$$

Where, M = mass
K = stiffness
B = damping coefficient
c = capacitance
v - applied voltage
x = deflection

The equation can be solved by using any number of system level simulators or ordinary differential equation (ODE) solvers. For the current project the software MATLAB/SIMULINK were used for this purpose. It should be noted that user must supply a suitable value for the Damping coefficient, B.

The simulation of the two models (Single Beam Resonator: Aluminum only and Composite, Beam Resonator: Aluminum and oxide) were performed for several states and the results are presented in Tables 1-3.

Planned Follow on Work

As an extension of the 1998 Summer Faculty Research Program (SFRP), it is planned that the software MEMCAD 4.0, SIMULINK and MATLAB will be used to develop MEMS models for RF switches. A summer research extension program (SREP) proposal for this project will be submitted shortly.

Table 1

Single Beam Resonator
Specifications: Aluminum Only
Gap Space: 1/um
Electrodes 89.7 um (3/4 of beam length)

Component	Applied Voltage (Volt)	Deflection (um) x	z
Rt elect	0.001		
Lft elect	0.2	5.86E-4 -	2.54E-4
Beam	40		
Rt elect	0.0001		
Lft elect	0.2	5.32E-4 -	2.54E-4
Beam	40		
Rt elect	0.001	1.1E-3 -	2.7E-4
Lft elect	0.3		
Beam	40		
Rt elect	0.001	1.9E-3 -	2.7E4
Lft elect	0.5		
Beam	40		
Rt elect	0.001	2.58E3 -	4.24E-4
Lft elect	0.5		
Beam	50		

Table 2

Single Beam Resonator
Specifications: Composite Beam (Aluminum)
Gap Space: 1 μm
Left Electrode Only: 89.7 μm

Component	Applied Voltage (Volt)	Deflection (μm) x	z
2 metal layers 1 metal layer Beam	0.1 0.001 60	0.1107	- 0.00548
2 metal layers 1 metal layer Beam	0.4 0.001 60	0.1390	- 0.00547
2 metal layers 1 metal layer Beam	0.002 0.001 60	0.1410	- 0.00548
2 metal layers 1 metal layer Beam	0.0002 0.0001 40	0.1410	- 0.00548
2 metal layers 1 metal layer Beam	0.0002 0.0001 70	0.2200	- 0.0081

Table 3A

Single Beam Resonator
Specification: Composite Beam (Aluminum and Oxide)
Beam: 15V
Left Electrode (Sense): 15V
Right Electrode (Drive): 0-15V (Variable)

Drive (Rt elect.) Volt	Deflection (μm) Max x
0	1.15E-02
3	7.38E-03
6	4.17E-03
9	1.89E-03
12	5.03E-04
15	4.09E-05

Table 3B

Single Beam Resonator
Specification: Composite Beam (Aluminum and Oxide
Beam: 8V
Left Electrode (Sense): 8V
Right Electrode (Drive): 0-8V (Variable)

Drive (Rt elect.) Volt	Deflection (μm) Max x
0	3.24E-03
2	1.83E-03
4	8.26E-04
6	2.16E-04
8	9.38E-07

Table 3C

Single Beam Resonator
Specification: Composite Beam (Aluminum and Oxide)
Beam: 5V
Left Electrode (Sense): 5V
Right Electrode (Drive): 0-5V (Variable)

Drive (Rt elect.) Volt	Deflection (μm) Max x
0	1.26E-03
1	8.12E-04
2	4.61E-04
3	2.09E-04
4	5.58E-05
5	4.54E-06

CFD-ACE+

CFD-ACE+ is an advanced computational environment developed by CFD Research Corporation (Ref. 3) for the simulation and analysis of fluid flow and associated physics in a wide variety of industrial applications. The software package adopted a numerical technique in which the solution space, or domain, is broken down into a large number of individual cells. The flow equations are integrated over the volume of each cell in a discrete manner. The values needed for the integration at the boundary of each cell are evaluated based upon neighboring cells and boundary conditions supplied by the user. When the integrated equations at each cell are collected, they form a matrix of equations that can be solved using iterative, numerical techniques.

The integrated CFD-ACE+ environment is comprised of three major modules:

1. CFD-GEOM
2. CFD-ACE/ACEU
3. CFD-VIEW

1. CFD-GEOM

The module generates geometry and grids for the models and interfaces with several commercially available CAD software packages such as Pro-Engineer, UniGraphics, AutoCAD, Plot 3D and IGES. The output formats of the module are suitable for solvers, such as Nastran, Patran, Plot3D, CFDRC Mixed, Fast and DTF.

2. CFD-ACE/ACEU

The module contains a number of multipurpose solvers for Structured and Unstructured grids. The modules can solve advanced models in the following application areas:

- flow (liquid or gas)
- heat transfer (including conjugate fluid/solid)
- mixing and reacting of multiple species
- radiation
- turbulence (low and high Re-models)
- reactions (instantaneous/equilibrium/finite-rate)
- spray (evaporating liquid/melting solid)

3. CFD-VIEW

The module permits interactive visualization and analysis of CFD data sets. The attributes include:

- Visualization objects (surfaces, point display of spray)
- Annotation objects (colormap legend, text labels, lines, arrows, boxes etc)
- Analysis tools and animation capability

Underlying all of the CFD-ACE+ modules is CFD-DTF or Data Transfer Facility, which provides a modular and expandable environment for transferring data.

The author participated in a formal training session for this software package offered by the CFDRC, Inc. and explored the possible applications of the package in MEMS (Microfluidic devices). However, no attempt was made to develop Microfluidic models using the CFD-ACE+ package, due to the limited period of the summer program.

Conclusion

Two of the commercially available Micro-electro-mechanical Systems (MEMS) CAD software were reviewed. The software MEMCAD 4.0 developed by Microcosm Technologies, Inc. was successfully used to develop two MEMS resonator models. The simulation results obtained from the models are very encouraging and it was concluded that further improvement of the simulation models are possible utilizing the capabilities of the software MEMCAD 4.0. Results from the Simulation models were used to study the behavior of the Macromodels using the software SIMULINK and MATLAB.

Acknowledgements

The author would like to acknowledge the support and encouragement given by Mr. Jim Collins, Chief, AFRL./IFTE Branch and Mr. Peter Rocci, focal point of the summer Research Project. Thanks are also due to Ms. Lee Lazicki for carefully typing this report.

References

1. "MEMCAD 4.0 Software Solutions for MEMS Design: User Guide". Microcosm Technologies, Inc., May 1998.

2. Swart, N.R., Bart, S.F., Zaman, M.H., Mariappan, M., Gilbert, J.R., and Murphy, D.. "Auto MM: Automatic Generation of Dynamic Macromodels for MEMS Devices." The 11th Annual International Workshop on Micro Electro Mechanical Systems. Proceedings of IEEE, January 25-29, 1998, Heidelberg, Germany.

3. "CFD-ACE+ (Version 4.0) Training Manual: AFRL, Rome NY," CFD Research Corporation, July 1998.

4. Rocci, P. and Das, D., "A Modeling and Simulation Approach for Micro-Electro-Mechanical Devices." Submitted for presentation at the Government Microcircuit Applications Conference, March 8-11, 1999, Monterey CA.

TOWARD AN ARCHITECTURE FOR A GLOBAL INFORMATION BASE

Venu Dasigi
Associate Professor
Department of Computer Science
Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, GA 30060-2896
vdasigi@spsu.edu

and

Paul O'Neil
Air Force Research Laboratory / IFED
Rome Research Site
32 Hangar Road
Rome, NY 13441-4114
oneilp@rl.af.mil

Final Report for:
Summer Faculty Research Program
Air Force Research Laboratory - Information Institute
Rome Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Air Force Research Laboratory - Information Institute
Rome Research Site

July, 1998

TOWARD AN ARCHITECTURE FOR A GLOBAL INFORMATION BASE

Venu Dasigi
Associate Professor
Department of Computer Science
Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, GA 30060-2896
vdasigi@spsu.edu

and

Paul O'Neil
Air Force Research Laboratory / IFED
Rome Research Site
32 Hangar Road
Rome, NY 13441-4114
oneilp@rl.af.mil

Abstract

A global information base is a comprehensive warehouse of information and a full suite of access services that satisfy the information needs of a variety of users in a timely manner. In the context of Air Force applications, the spectrum of users may span the warfighter through the theater commander and chief of staff and the legislators to the president. The information base would involve a variety of media, such as text, speech, audio, image, video, etc., be stored in multiple repositories, and be accessible from many different locations. It is also expected to be dynamic in the sense that the information base organizes itself over time to satisfy the information needs of users.

The short time span of this project forces us to limit the scope of this work to the textual medium, but we expect that some of the ideas apply to other media, as well. In arriving at a better understanding of the vision of the global information base leading to an architecture, we follow Booch's macro development process. We first conceptualize the vision, analyze the desired system behavior through scenarios, and arrive at an architecture, keeping the current state of the art in mind.

TOWARD AN ARCHITECTURE FOR A GLOBAL INFORMATION BASE

Venu Dasigi *and* Paul O'Neil

1. Introduction

The USAF Scientific advisory board prepared a document entitled, "New World Vistas," in which the different panels of the board made recommendations for future technological investments. According to this comprehensive set of documents: *Information is the capital commodity of the future* [NWV-IT, 95]... *The Air Force must expand its traditional role as the leading proponent of air and space power to include an equally important role in cyberspace* [NWV-S, 95]... *Both micro wars and major regional contingencies have become information intensive conflicts. As a corollary, warfare will emerge* within *the information domain driven by the proliferation of computer technology, low cost of entry, and large numbers of attractive military and civilian targets* [NWV-IA, 95].

A *Global Information Base* (GIB) project is envisioned to improve global awareness, leading to information dominance. Briefly, the purpose of this project is to support information warehousing, storage and retrieval to provide a full suite of access services for on-time C^4I (Command, Control, Communications, Computers, and Intelligence) information to the warfighter, and to develop techniques for comprehensive knowledge of the battlespace so that all warfighters, from the theater commander to each individual combatant has the proper information to support their decisions and actions. Thus, the scope of a GIB is very broad. Sources of information that goes into the information base, as well as the information base itself are heterogeneous (both in terms of media and formats) and geographically dispersed.

In this report, we examine the vision of a global information base, and arrive at a first cut architectural design by following Booch's macro development process, as outlined in [Booch, 94]. The major principles guiding the change architecture are: to identify components that are unlikely to change which form the basis for the design, to identify those that are likely to change so as to ensure that the architecture enables such changes, and traceability. The latter refers to the ability to match functional specifications to design (to ensure completeness) and vice versa (to avoid redundancy).

To circumscribe the scope of this work, we limit ourselves to the text aspects of the GIB. We do believe, however, that there are some ideas in here that apply to other media as well. The rest of this report is organized as follows. The next section on "Conceptualization" establishes the vision and general goals, and clarifies any assumptions. In the section on "Analysis," we provide a model of the desired behavior of the GIB. We do this by first identifying a general functional specification, and then elaborating different "scenarios" involving related behaviors. Thus, the section on "Analysis" provides the users' perspective. The following section, "Architecture," identifies the different functional components, and describes how they interact to support the desired functionality. This architecture would be the starting point for a more detailed design, and we identify some of these details. The last section, entitled "Conclusion," summarizes the main points, identifies some issues, and outlines future opportunities to continue this work.

2. Conceptualization

It is envisioned that the global information base (GIB) is a

- decentralized,
- geographically dispersed,
- dynamic, and
- self-organizing

store of information that is heterogeneous in terms of

- media (text, speech, audio, image, video, signal, etc.) and
- formats (structured, unstructured, formatted, unformatted, etc.)

with

- multiple data generators and
- multiple repositories,

and supporting users such as

- the president,
- legislators,
- theater commander,
- individual combatants, et al.,

that provides a full suite of access services for

- decision support and
- transaction processing,

supporting multiple levels of

- security and
- information need,

maintaining

- speed,
- currency, and
- precision

of information.

The dynamic nature of the GIB makes it grow and evolve continuously. The GIB is also self-organizing, which means that each incoming document results in a modification of the information base and also that a number of demon-like[1] modules watch for incoming information and interpret it for different purposes, providing useful views to users. Although the information in the GIB spans different media, we focus now on just the text portion, which can be structured (making it possible to store some of the information in a database for efficient access) or in an unstructured, free text form. The presence of multiple data generators and repositories makes it an information "warehouse," making many issues of data warehousing relevant to this project [Widom, 95].

The presence of different categories of users means that some have extremely real-time information needs, while others have less urgent needs. In some categories there are many more users than in others, and users belong to different priority levels, too. Further, the broad range of users necessitates multiple levels of security, as well. The spectrum of access services is broad and will be dealt with in the next section.

The following general assumptions are made to support the vision of the GIB as outlined above:

- Currency of information is critical for most users, especially when speed is also critical.

[1] See [Winston, 92] and [Winston and Horn, 88].

- Speed is of the essence for some users; if a tradeoff between speed and other parameters, such as storage space, arises, unless there is specific justification otherwise, speed takes precedence. However, there are categories of users for whom speed may not be THE critical factor.
- In order that the GIB adapt itself to the users, the categories of users to whom speed is not critical may be asked for a quick feedback on the access services.
- It is important to distinguish information (contained *in* a document) and meta-information (which is *about* the document itself); the latter may sometimes be contained in the document.

3. Analysis

There are two kinds of functionality supported in the GIB. The first category of functions are those initiated by users to satisfy their information needs. The second set of functions are not directly initiated by users, but are related to the maintenance of the GIB. It is expected that such maintenance will be automatic in that as new documents enter the GIB, different actions are set in motion. Below we list these two categories of functionality, and then identify a few scenarios that might arise as the GIB is used, so interaction between the different functions (and consequently that between the design modules responsible for those functions) becomes clear.

3.1. Functionality related to Usage

- Retrieval based on keywords (key phrases) and index words / phrases
- Conceptual retrieval using open-ended descriptions
- Database retrieval (based on structured information such as dates, people, places, etc.)
- Browsing closely related items
- Retrieve a summary / abstract, or a set of keywords (key phrases) for a document
- Give relevance feedback
- Look up words / phrases conceptually related to a given word / phrase
- Natural language question - answering

- Cross-language and cross-media retrieval (retrieval of related information that is possibly in a different language or a different medium, e.g., an image related to the topic at hand)
- Create / edit user profile
- Filter / route information based on profiles (although this function is initiated by each incoming document, rather than by the user)

3.2. Functionality related to Maintenance

- Automatic indexing (date, keyword, etc.)
- Automatic hyperlinking (facilitates a kind of browsing)
- Meta Database generation
- Data mining (based on information stored in the database)
- Summarize / abstract documents
- Automatic document classification
- Maintain / update user profiles
- Communicate with other repositories of the warehouse

3.3. Scenarios

The "system" we refer to below in scenarios 1, 2, and 3 is the GIB server that links users to the GIB, which itself is distributed into many, possibly "active," repositories. The word "active" here simply means that some local processing specific to the repository takes place there. In scenario 4, the "system" refers to the local processing at each active repository.

3.3.1. Scenario 1 - User-Initiated Retrieval

a. User: Attempts to retrieve information based on a query by specifying either keywords / index words or a more open-ended descriptor.
b. System: Processes the query locally (that is, at the GIB server) and communicates with all repositories in each of the steps that refer to the system below.

c. System: For Efficiency, the query is processed first through a database retrieval process. (This may be possible if the retrieval is based on the information stored in the databases, but not always.)
d. System: Instead of / In addition to the above, one or more information retrieval algorithms (e.g., vector space [Salton, 88], LSI-based [Deerwester, et al., 90], machine learning) are invoked on the query, too.
e. System: Fuses the results from all repositories together (e.g., [Towell, et al., 95]) and presents to user.
f. User: Gets a list of items displaying the header information.
g. User: Has several choices now:
    i. either selects some documents and asks for a summary or full text,
    ii. or browses similar / related documents first and then asks for a summary or full text,
    iii. or chooses to query by example.
h. System: Before quitting, asks user to rate items (this can be done implicitly or explicitly), if the algorithm is trainable.
i. User: Responds.
j. System: Uses the rating information to train the algorithm.

3.3.2. Scenario 2 - Information Filtering / Routing

In this scenario, the system starts with a set of user profiles / preferences (either matched to the classification scheme, keywords, or a customized filtering algorithm [Lang, 95]), which evolve over time, based on the relevance feedback a user provides. The user may be asked to set some initial preferences when (s)he wants to use the "filtering / routing" service. Each repository has a classifier that classifies documents as they enter into the GIB and a collection of new documents, that is, documents that have not been processed by at least one user profile and not timed out.

a. System: Either through profile matching or a filtering algorithm, routes the document to interested users either when they access the GIB or during off-line processing.
b. System: Processing in step a is performed locally (that is, at the GIB server) while communicating with all repositories. This procedure involves combining results from all the individual repositories.
c. User: Gets a list of items displaying the header information.

d. User: Has several choices now:
   i. either selects some documents and asks for a summary or full text,
   ii. or browses similar / related documents first and then asks for a summary or full text,
   iii. or chooses to query by example.
e. System: Before quitting, asks user to rate items (this can be done implicitly or explicitly).
f. User: Responds.
g. System: Uses the rating to update the user profile.

3.3.3. Scenario 3 - Natural Language Question-Answering

a. User: Makes a query in natural language.
b. System: Parses the query appropriately.
c. System: Ascertains from the user if
   i. a list of documents is needed or
   ii. a direct natural language answer is needed.
d. System: The request is processed by the natural language processor (NLP) locally (that is, at the GIB server) while communicating with all repositories. This procedure involves fusing results from all the individual repositories.
e. System: Responds to the user depending on the case:
   i. In this case, simply gives the user a list of items, as in step e of scenario 1. The rest of the scenario is similar to the steps that follow in scenario 1.
   ii. In this case, performs further processing to extract and present the answer to the user from the top-ranked documents (this step possibly involves, among others, data mining algorithms).
f. User: Responds depending on the case:
   i. In this case, the rest of the scenario closely follows from step f of scenario 1.
   ii. In this case, the user either rephrases the question, or asks a new question (either way the process repeats from step a), or simply stops.

3.3.4. Scenario 4 - At the Repositories

The preceding scenarios deal with the user's perspective. So, the "system" mentioned in those scenarios is the GIB server the users directly deal with. This scenario refers to what happens as a document enters (or is entered) into the GIB, and consequently in this scenario, "system" refers to the local GIB processor. Since there are multiple repositories, the scenario repeats itself at *each* repository.

a. External world / Maintainer: Causes a new document to enter the GIB.
b. System: Enters the document into the new document set. (Conceptually, it is removed from this set and into the main storage if either it times out or if all user profiles have processed it.)
c. System: Enters the document's (possibly partial) contents into the databases or the meta database, if the document is structured enough, or if it has some structure that can be exploited. (This step is important for efficiency.)
d. System: Classifies the document into the classification scheme.
e. System: Indexes the document (by date and / or keywords).
f. System: summarizes / abstracts the document (this can be done on demand, but in the interest of speed, this is done as the document enters the GIB).
g. System: Extracts information such as dates, cities, places, people, and indexes them to support data mining during natural language question-answering.

4. Architecture

As we identified the detailed scenarios in the preceding section, several important elements of the architecture became obvious. First, there are many users and many repositories. There are different choices to establish the necessary interactions. In the choices we made, we were guided by considerations of speed, timeliness, and consistency (e.g., store user profile information in a single module to minimize communication costs). Obviously, there are several repositories, and any processing on the data / information at a repository must be local. It is not only convenient to have a single GIB server with which users can communicate, but it is consistent with the rest of the preceding considerations. Thus, we opted to have a single GIB server and multiple repository servers, with individual users communicating with the GIB server through thin-clients[2], resulting in a layered architecture. See Figure 1. We now describe the architecture.

[2] See [Kanter, 98] for an exposition to thin-client / server computing.

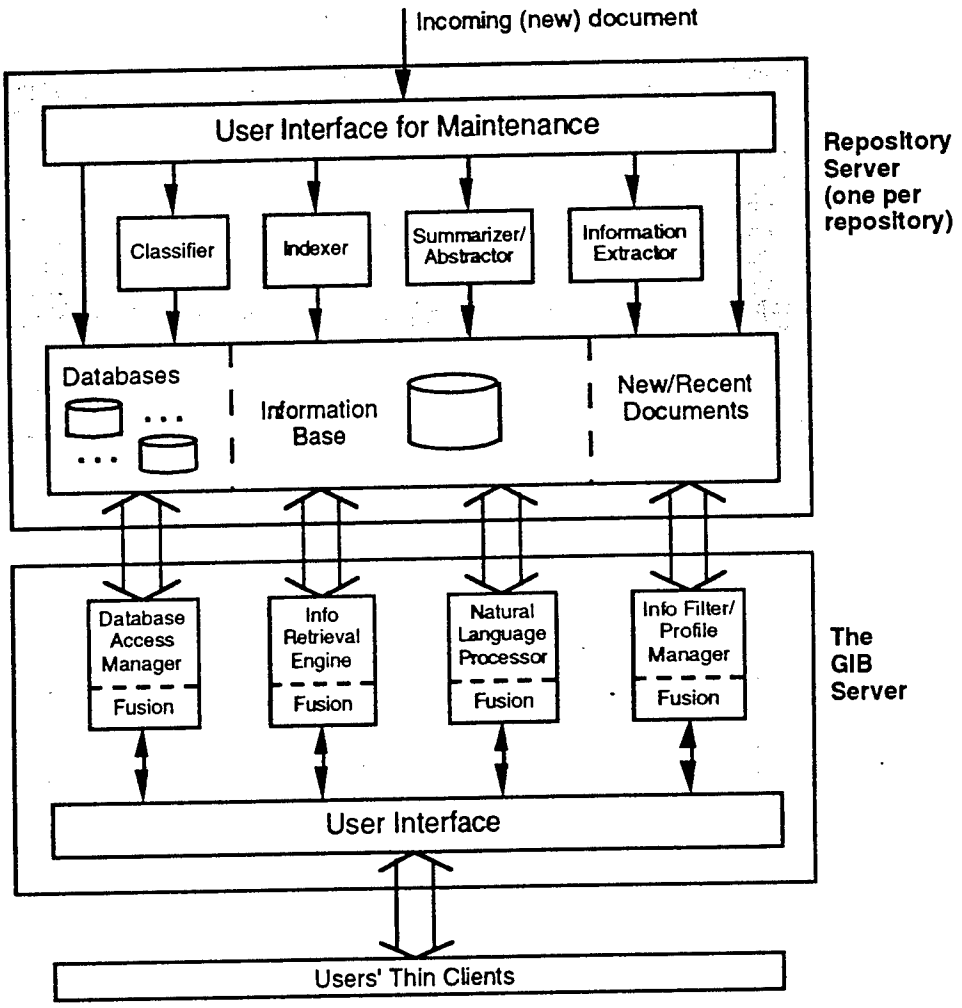


Figure 1. The Global Information Base Architecture

Users communicate with the GIB server through their *Thin-clients* [Kanter, 98]. A thin-client is a client which does not do much processing. Thin-client server computing supports high speed, different levels of data security, hardware independence, full windows-based interface, and load balancing (if multiple servers are involved). Thus, this model seems to be ideally suited for user communication with the GIB, considering that users are very heterogeneous in their needs, security levels, and their hardware.

The *GIB server* forms the layer between users and the GIB repositories, providing a uniform interface to the users. In communicating with individual users, it supports a good user interface. It also has modules dealing with different tasks: a *Database (DB) access manager*

(DBAM), an *Information retrieval (IR) engine (IRE)*, a *Natural language (NL) processor (NLP)*, and an *Information filter (IF) / Profile manager (PM)*. Since all these modules communicate with all repositories, they incorporate fusion methods to combine the information retrieved from the repositories and present a unified picture to the user. These fusion methods range from a simple set union to much more complex statistical and semantic methods.

There is a *Repository server* (RS) corresponding to each information repository. An RS includes multiple databases, an information base, and a number of modules that process data as they enter the repository. A *Classifier*, an *Indexer*, a *Summarizer / Abstractor*, and an *Information extractor (IE)* process documents as they enter the repository. Any structures these modules need for their processing are also stored at the repository server.

In short, any data related to users is stored at the GIB server, without duplication (and the associated problems of redundancy and update consistency) at the repositories. Further, any knowledge sources that the GIB server modules need (e.g., a thesaurus - more on them below) are stored just at the GIB server. Any data / information associated with individual documents (e.g., an index - more on them below) is replicated at each repository, yet with no redundancy.

Now we zoom into the two major layers. Some details of the GIB server are shown in Figure 2. The GIB server has a *User Interface (UI)*, *Database access manager (DBAM)*, an *Information retrieval engine (IRE)*, an *Natural language processor (NLP)*, and an *Information filter (IF) / Profile manager (PM)*. The user interface may support windows-based interaction, either dialog-based or menu-based. While the physical databases are stored at the repository servers, storing all the data dictionaries at the DBAM may be appropriate. This would of course mean that any changes to the database organization at any RS need to be coordinated with the DBAM at the GIB server. The DBAM has an SQL generator to process users' request that can be handled by the DBAM (as opposed to requiring the IRE). The SQL generator would first have to check the request to make sure it is appropriate for the DBAM. For reasons described before, each major module in the GIB server has a fusion component.

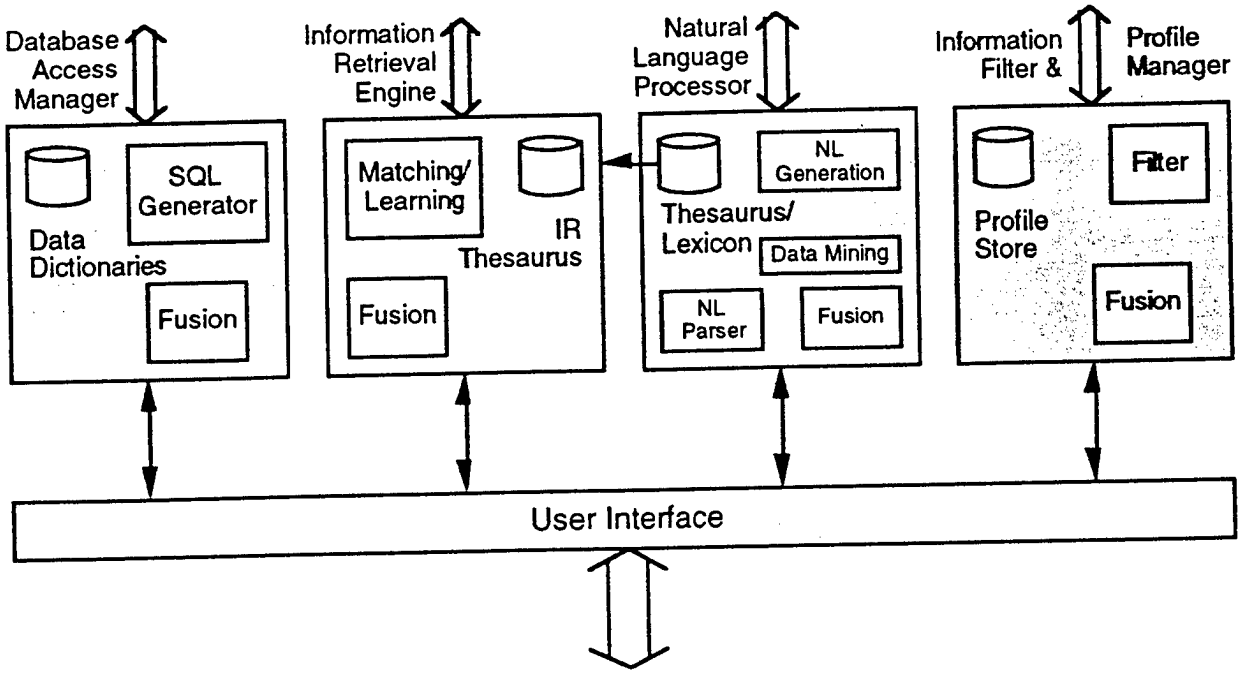


Figure 2. The Global Information Base Server

The IR engine incorporates statistical, vector, and machine learning based retrieval algorithms for matching an information need with relevant information stored in the information base. Details such as stop word filtering, stemming, etc. are not shown. Thesaurus information may be distributed between the GIB server and the repositories. The IR engine exploits many structures associated with the information base, e.g., indexes, classification, etc. The NL processor has its own lexicon and thesaurus to support open-ended language processing. It contains modules for parsing NL input, and for generating appropriate answers in NL, as well as data mining algorithms. Many contemporary techniques relevant to the IR engine and the NL processor are covered in [Jacobs, 92]. The Information filter / profile manager has a global profile store. The filter performs a matching between a user profile and recent documents when the user accesses the GIB, if speed is not a critical factor. For higher priority users, such matching is performed off-line, not necessarily when the user accesses the GIB. Either way, an index of recent documents relevant to the user is saved along with the user profile.

Each repository server (see Figure 3) has several *Databases*, an *Information base* (by this we refer to all unstructured information from the documents), a *Classifier*, an *Indexer*, a *Summarizer / Abstractor*, and an *Information extractor*. There is a meta database storing data about

the documents in the information base, and other databases that store any structured information input into the repository. The meta database stores, in each record, general information such as date, source, etc. along with a pointer to the unstructured (full text) document in the information base. The information base itself stores all the documents and is linked to the new / recent document storage area. A thesaurus derived from the documents in the information base is often useful to retrieval and classification algorithms. The classifier, possibly using the thesaurus, associates each document in the information base to zero or more classes in the classification scheme, and this information is stored as classification information. The indexer creates one or more indexes on the information stored in the databases and the information base. The summarizer / abstractor creates summaries / abstracts of the documents using natural language techniques, and may be thought of as a specialized natural language processor. As such it uses NL parsing techniques and a lexicon (not shown). The information extractor extracts key concepts such as dates, people, places, etc. and indexes the documents to the extracted information that can be accessed for data mining.

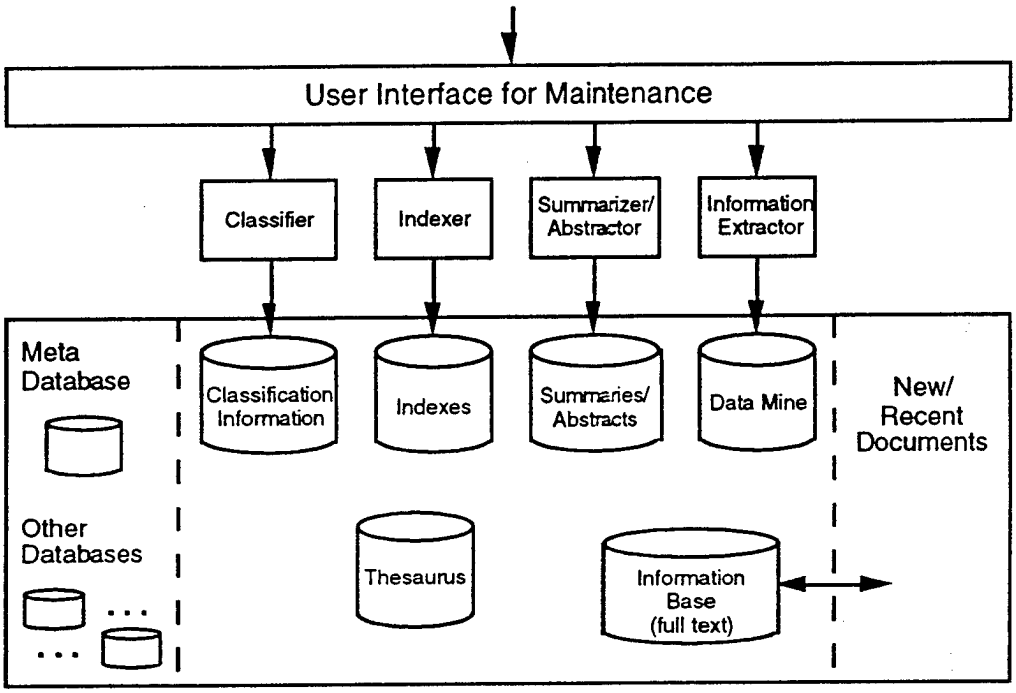


Figure 3. A Repository Server (one per repository)

5. Conclusion

This work represents a first attempt at identifying an appropriate architecture to support a global information base, focusing mainly on the medium of text. Besides providing a high level architecture for the GIB, we hope that this document contains enough details to flesh out a detailed design of the individual components. Development of some of the components involves substantial research effort, but past and recent research results support the development of many of the modules. It is best to view the GIB project as comprising several smaller projects. We believe the architectural design developed in this work has helped identify these smaller projects. The architecture is based on the assumption that timeliness and speed of information access are critical to the central purpose of the GIB.

In summary, we have a layered architecture with several individual components. At the heart of the architecture are the GIB server and the individual GIB repositories. The GIB server has a user interface that communicates with individual users through thin clients. Individual modules for database access, retrieval, NLP, and information filtering and profile maintenance are also part of the GIB server. Each of these modules necessarily involves one or more fusion algorithms to reconcile the information flowing into them from the individual repositories. Each GIB repository has a very large store comprising a local information base and several databases. They also contain modules that process documents as they enter the GIB: a classifier, an indexer, a summarizer, an information extractor, etc., and possibly (although not necessarily) a user interface for local maintenance.

References

1. [Booch, 94] Booch, Grady: Object-Oriented Analysis and Design with Applications, Second edition, Benjamin / Cummings, 1994.
2. [Deerwester, et al., 90] Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman: Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, 41(6), pp. 391-407, 1990.
3. [Jacobs, 92] Jacobs, Paul: Text-Based Intelligent Systems - Current Research and Practice in Information Extraction and Retrieval, Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.
4. [Kanter, 98] Kanter, Joel: Understanding Thin-Client / Server Computing, Microsoft Press, 1998.
5. [Lang, 95] Lang, Ken: Newsweeder: Learning to Filter Netnews, *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331-339, 1995.

6. [NWV-IA, 95] New World Vistas - Air and Space Power for the 21st Century, Information Applications Volume, USAF Scientific Advisory Board, 1995.
7. [NWV-IT, 95] New World Vistas - Air and Space Power for the 21st Century, Information Technology Volume, USAF Scientific Advisory Board, 1995.
8. [NWV-S, 95] New World Vistas - Air and Space Power for the 21st Century, Summary Volume, USAF Scientific Advisory Board, 1995.
9. [Salton, 88] Salton, Gerard: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1988.
10. [Towell, et al., 95] Towell, G., E. Vorhees, N. Gupta, B. Johnson-Laird: Learning Collection Fusion Strategies for Information Retrieval, *Proceedings of Twelfth Annual Machine Learning Conference*, 1995.
11. [Widom, 95] Widom, Jennifer: Research Problems in Data Warehousing, *Proceedings of 4th International Conference on Information and Knowledge Management (CIKM)*, 1995.
12. [Winston and Horn, 88] Winston, Patrick and Berthold Horn: LISP, Third edition, Addison-Wesley, 1988.
13. [Winston, 92] Winston, Patrick: Artificial Intelligence, Third edition, Addison-Wesley, 1992.

Amplitude and Frequency Modulation Characteristics of Stressed Speech

Kaliappan Gopalan
Professor
Department of Engineering

Purdue University Calumet
Hammond, IN 46323

Final Report for
Summer Faculty Research Program
Rome Research Site

Sponsored by
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Research Site

July 1998

Amplitude and Frequency Modulation Characteristics of Stressed Speech

Kaliappan Gopalan
Professor
Department of Engineering
Purdue University Calumet
Hammond, IN 46323

Abstract

An analysis of stressed speech using its amplitude and frequency modulation (AM and FM) characteristics was carried out. From the preliminary results using actual stressed speech from an operational aviation emergency, stress, in general, appears to increase the modulation behavior in both AM and FM. In particular, the modulation characteristics become significantly more if the analysis (carrier) frequency is centered at one of the resonant frequencies of the vocal tract. Moreover, the spectra of AM envelopes show increasing peak frequencies with stress. The highest peak in the spectrum of the demodulated instantaneous frequency as well as in the AM envelope spectrum, appears to track the fundamental frequency F0. Linear prediction model of the envelope also shows a peak at F0 regardless of the analysis (center) frequency; the peak may not be distinct all center frequencies, however. These features can be used to detect the presence of stress in a speaker.

Amplitude and Frequency Modulation Characteristics of Stressed Speech

Kaliappan Gopalan

Introduction

Efficient electrical transmission of information commonly involves modulation of amplitude or frequency, of a carrier signal. In human speech communication, modulation not only carries the spoken message in a natural way but also conveys other information such as the speaker's emotional and physiological state and personality. Although modulation may be discarded for message perception by humans, detection of the speaker's stress has many applications. Speech and speaker recognition under adverse, stressful environments, for example, is required for accurate response by emergency management personnel. Other applications include workload modification based on physiological stress, detection of a speaker's deception for law enforcement use based on emotional state, and implementation of natural-sounding speech synthesizers.

This report discusses a brief overview of current research on speech features used to detect stress, and presents the observations made in the study of the modulation characteristics of speech from two pilots in an emergency.

Acoustic Correlates of Stresses Speech

Studies have shown that human speech has emotion and other nonlinguistic information encoded in it. Increased activation of the sympathetic nervous system or the parasympathetic nervous system is observed to occur when a speaker is angry, fearful, sad, etc [1]. This increased activation leads to change in heart rate and blood pressure, and also to tremor in muscle activity. Consequently, the articulatory and respiratory movements for speech production are affected. The spectral characteristics of the resulting "stressed" speech are, in general, observed to have increased fundamental frequency F0, increased amplitude, and decreased speech duration. The extent of variation in F0, however, has been shown to depend on the speaker and the type of stress. Other manifestations of stress in speech include variations in the formants and their bandwidths, increase in high frequency energy, and changes in the glottal pulse shape.

Many studies have been conducted to analyze the acoustic correlates of stress using simulated and actual speech data. The following brief description summarizes some of the more recent work. Using laboratory color-word test (French version) and the utterances from the cockpit voice recorder of a crashed aircraft, Absil, et al [2], for example, concluded that the pitch-based measures of stress were able to discriminate different degrees of stress. They found that at very high stress levels (as when the aircraft was about to crash), the fundamental frequency itself was very high and more efficient in identifying stress. By examining the speech between an F-16 pilot and his wingman during an emergency, Benson [3] observed that the pitch and the first

formant were higher than normal in stressed speech. A further correlation between changes in F0 and perceived stress in speech was found by Protopapas and Lieberman [4]. By changing the mean and maximum F0 within an utterance in normal speech, these researchers found that listeners' rating of stress increased with increased values of F0. However, aside from physiological and emotional stress, Junqua observed that Lombard speech also increases F0 [5]. In addition, Lombard speech mimics stress conditions [6] in increasing speech amplitude, vowel durations, formant frequencies, etc. Therefore, an analysis of other characteristics of speech is needed in addition to the variabilities in the spectral characteristics to correlate stress.

Motivation for the Study of Speech Modulations

Some of the important characteristics of speech signal are evident from the resonances of the vocal tract. The analysis of the resonant frequencies (formants) and their bandwidths bring out the multicomponent structure of the speech signal. In addition, studies have shown that in stressed speech, the fundamental frequency of excitation and formants have higher variations in their instantaneous values than in neutral speech. Therefore, an analysis of the instantaneous frequency variation around F0 and the formants is believed to bring out the frequency variations due to stress. In addition to the frequency modulation (FM) causing instantaneous frequency variations, Teager and Teager [7] have shown that speech resonances also have time-varying amplitudes. Thus, a study of some of the nonlinear mechanisms of speech production via amplitude and frequency modulations may be better able to bring out the stress-related speech parameters. Based on this premise, Zhou, et al [8] recently used frame-based FM variation for stress classification and assessment. Using different vowel sounds extracted from the pilot of an F-16 aircraft in emergency, they concluded that the FM variation as a feature was less reliable than the pitch variation or the autocorrelation envelope area of the energy profile. They analyzed only the instantaneous frequency variation around the pitch of each utterance, however.

In the present work both amplitude envelope and instantaneous frequency variations from the AM and FM characteristics of speech were studied at different center frequencies to determine their ability in characterizing stressed speech.

Algorithm used for obtaining the the AM and FM Characteristics of Speech

It has been shown that the bandwidth of each of the formants in a speech signal is a product of both a FM component and an AM component. With multiple resonances of the vocal tract (formants), this model may be represented by [9]

$$x(t) = \sum_{k=1}^{N} A_k(t)\cos\left(\omega_k t + \Phi_k(t)\right) \qquad (1)$$

where $A_k(t)$ is the amplitude envelope (amplitude-modulating function, AE) of the k^{th} formant at frequency ω_k, and Φ_k is the instantaneous frequency (frequency -modulating function, IF) of the k^{th} formant out of N measurable formants. Because of the unlimited number of combinations of AM and FM that can give rise to the modulated signal at each formant, the analysis must be carried out separately at each formant to track its IF, bandwidth and AE. Representing the discrete-time version of a speech signal band-pass filtered at the center frequency f_c by

$$\dot{s}(n) = a(n)\cos\Phi(n)$$

or,

$$s(n) = a(n)\cos(\Omega_c n + \Omega_m \int_0^n q(k)dk), \tag{2}$$

the instantaneous frequency Ω_i given by

$$\Omega_i(n) = \frac{d\Phi}{dn}(n) = \Omega_c + \Omega_m q(n) \tag{3}$$

Applying Kaiser's algorithm [10] to obtain the Teager energy profile of the bandpass-filtered speech signal s(n) at frequency $f_c = \Omega_c/2\pi$ as

$$\psi[s(n)] = s^2(n) - s(n-1)s(n+1), \tag{4}$$

the instantaneous frequency (IF) Ω_i and the amplitude envelope (AE) are given by [11,12]

$$\Omega_i(n) \approx \arcsin\sqrt{\frac{\psi[s(n+1) - s(n-1)]}{4\psi[s(n)]}} \tag{5}$$

$$|a(n)| \approx \frac{2\psi[s(n)]}{\sqrt{\psi[s(n+1)] - \psi[s(n-1)]}} \tag{6}$$

Stressed Speech Database and the AM and FM Features Studied

The NATO database SUSC-0 was used to study the AM-FM characteristics of stressed speech. This database consists of approximately 10 minutes of interchange between an F-16 pilot, his wingman, and the tower in an in-flight emergency. During the course of flight, the pilot notices his hydraulic oil pressure dropping, realizes the loss of his engine, declares 'mayday' emergency, and, with the help of his wingman and the tower, lands the aircraft safely. Although the pilot's voice during the course of emergency was remarkably calm, one can assess the stress in this life-threatening situation.

For analysis, the utterance */is/* by the pilot was extracted from his speech during (a) his checking of the instrument panel, (b) his noticing of oil pressure being low, (c) his declaration of emergency, (d) his declaration of 'mayday', and (e) his urgent request for help in landing. The context for the utterance */is/* with its approximate time marking in the database is shown in Table I for each of the five cases to indicate the level of stress

Table I
Utterance context for pilot viper3

No.	File name	Utterance context	Approximate time marker in the database	Number of samples
1	vi3-is-25	oil pressure *is* about ..	25 s	7781
2	vi3-is-249	... know what an sfo *is* ...	249 s	3322
3	vi3-is-415	... emergency zero four *is* ...	415 s	1780
4	vi3-is-440	... it *is* ... lost my engine ... mayday mayday mayday	440 s	3327
5	vi3-is-450	... *is* an emergency I'm engine out	450	2125

Initially, modulations in the above utterances were analyzed at the fundamental frequency, F0. Since stress is known to affect F0, and also since F0 varies within an utterance, the 'nominal' fundamental frequency of the first utterance, vi3-is-25, was used in the analysis of all of the five utterances shown in Table I. Using the Entropic 'formant' function, F0 was obtained with a window duration of 15 ms with a 10 ms overlap. From the results of the formant function, F0 = 90 Hz was used. Note that this value was picked as an approximation to the most commonly occurred frequency for the 'neutral' or 'unstressed' utterance; it is neither the average nor the median value of the fundamental frequency for the utterance */is/* by the pilot.

To analyze the modulation behavior around F0 = 90 Hz, each utterance was bandpass-filtered using the discretized Morlet wavelet function,

$$h(t) = e^{-Kt^2} \cos(2\pi f_c t) \tag{7}$$

with f_c = F0 = 90 Hz, K = 10,000, and 601 samples with a sampling interval of 1/16000 s, (the same as that of the speech data). After bandpass filtering, the Teager energy profiles of the signal s(n) and its advanced and delayed versions, s(n+1) and s(n-1) were evaluated. Since the bandwidth of these profiles is much smaller than the signal bandwidth, the profiles were smoothed to reduce the effect of noise. From the smoothed energy profiles, the AE and the IF were obtained using Eqs. (5) and (6). Fig. 1 shows (a) the filter characteristics, (b) the spectrum of the signal and its filtered version, and (c) the AM AE and the FM IF variation in the filtered signal for vi3-is-249.

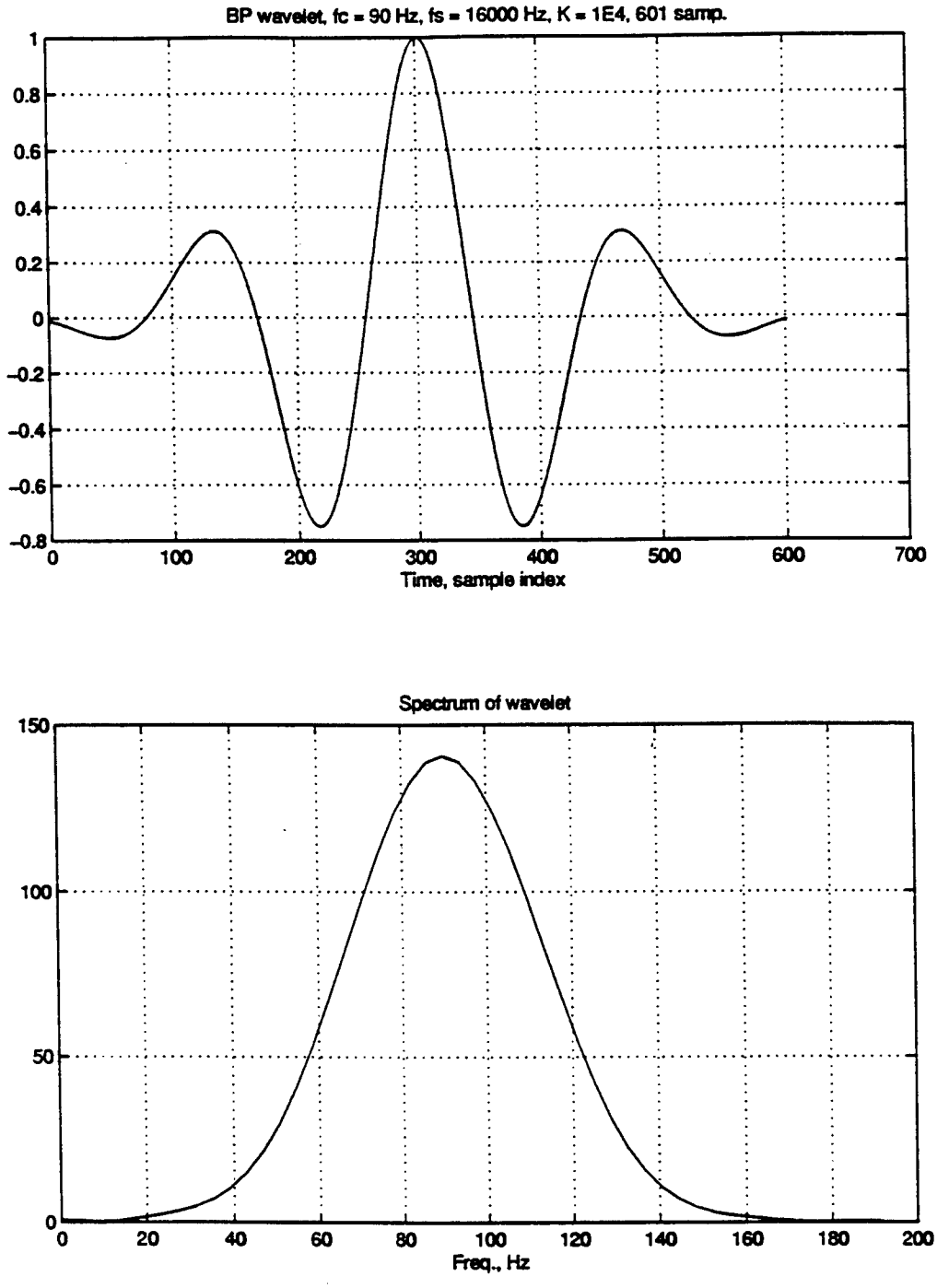
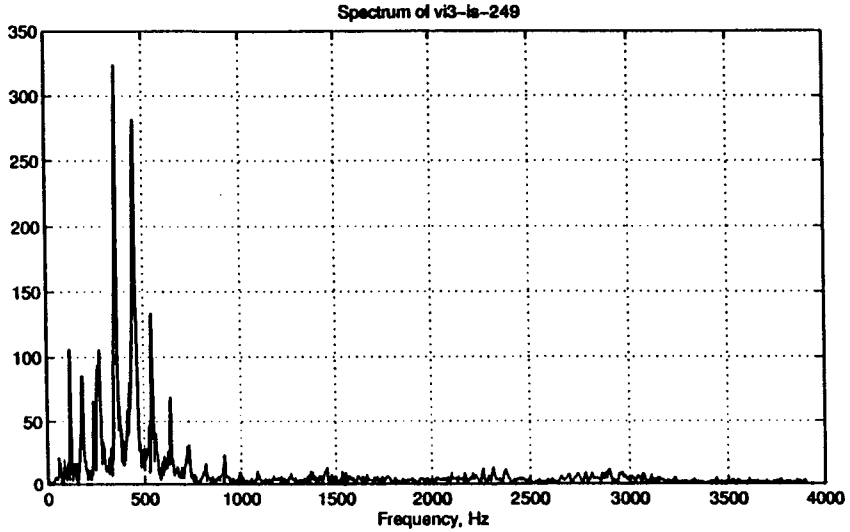
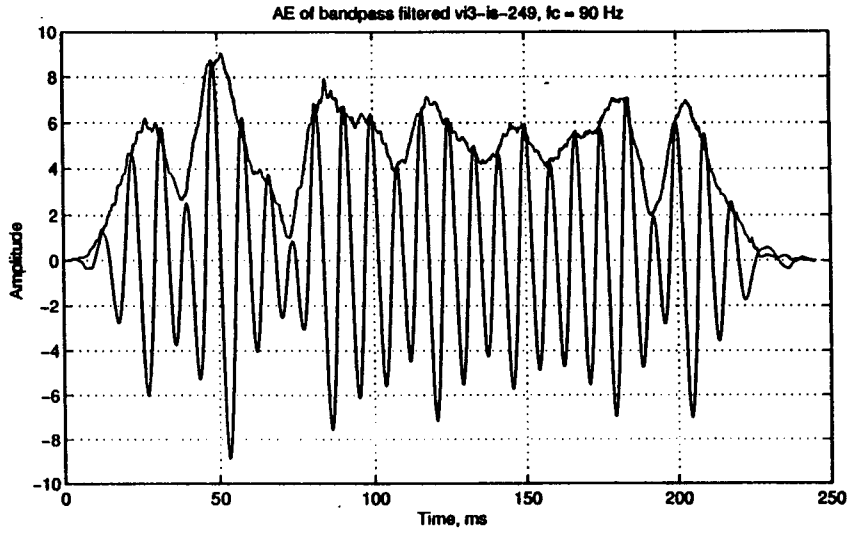
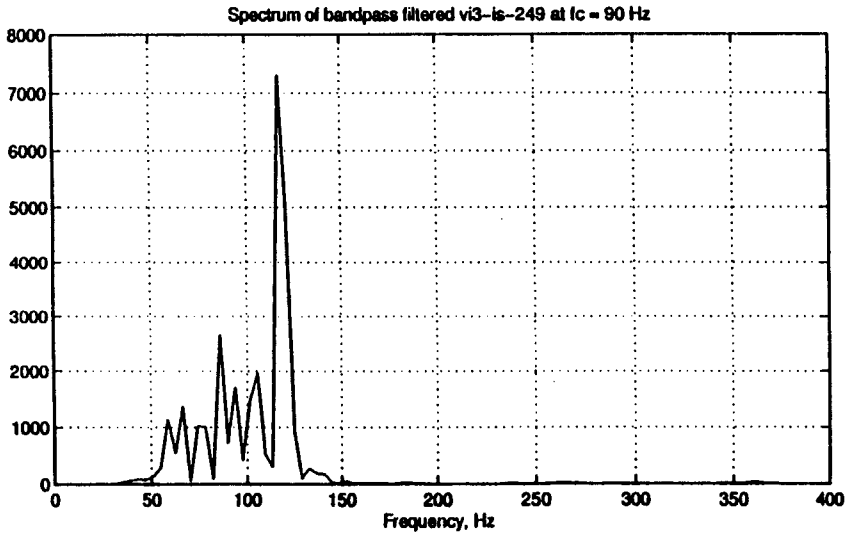


Fig. 1 Modulation characteristics of vi3-is-249 (neutral or low-stressed speech) around 90 Hz.

(a) Characteristics of bandpass filter used







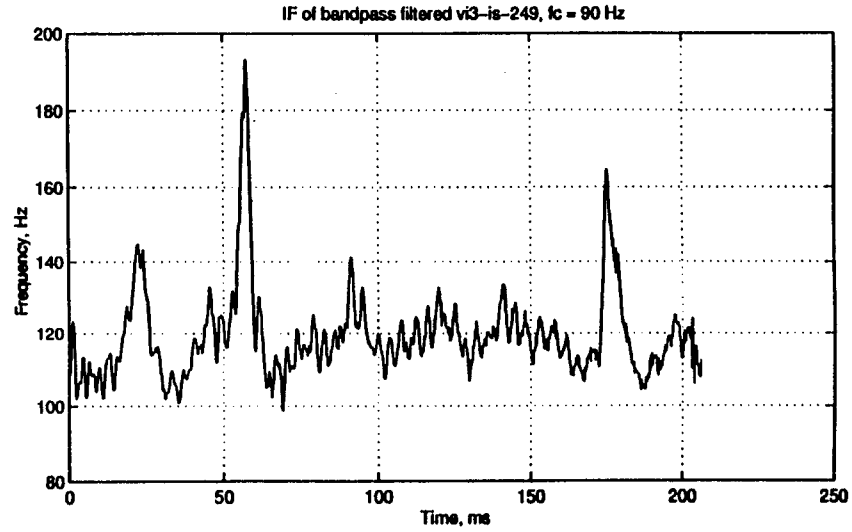


Fig. 1 (cont.) (b) spectrum of the signal and its filtered version, (c) AM AE (with the signal superimposed) and FM IF in the filtered signal

To compare the modulation characteristics of the utterances at different instants of stress levels, two ratios were defined. The first ratio R considers the energy (squared magnitude) of the AM envelope with that of the band-pass filtered signal. Clearly, at higher levels of amplitude modulation, R would be higher. The second ratio, flatness ratio f_a, relates the variation of the discrete envelope a = $[a_1\ a_2 \ldots a_N]$ to a constant value; it is defined by

f_a = geometric mean of the envelope, m_g/arithmetic mean of the envelope, m_a

where $m_g = \sqrt[1/N]{\prod_{k=1}^{N} a_k}$, and $m_a = \left(\frac{1}{N}\right)\sum_{k=1}^{N} a_k$

Clearly, the maximum possible value for f_a is unity, which occurs when AE is constant (flat). (Note that to avoid a ratio of zero because m_g becomes zero if the envelope starts or ends in zero, only the nonzero samples in the middle of AE are used to compute both the means.)

In addition to the two ratios, the spectral behavior of the AE and the IF were studied. It has been observed that the high frequency energy in speech increases with increasing heart rate. Inasmuch as stress affects the heart rate of the speaker, modulation characteristics at frequencies higher than the fundamental were also studied. The features observed for stressed and unstressed speech at various frequencies are discussed in the next section.

Observed AM-FM Features and Their Correlation to Stress

Demodulation in the vicinity of F0: For the pilot Viper3, the AM envelopes for the utterances in files 1 to 5 given in Table I showed, in general, an increasing amplitude with stress. The instantaneous frequency of the FM also showed an increase in jitter with stress. The ratios R and f_a based on AM and FM modulations around 90 Hz are shown in Table II.

Table II

R and f_a values for Viper3 around f_c = F0 ≈ 90 Hz

File name	R	f_a
vi3-is-25	1.9048	0.7882
vi3-is-249	1.9145	0.6423
vi3-is-415	1.7524	0.6867
vi3-is-440	1.9462	0.7320
vi3-is-450	1.8669	0.6858

As seen from Table II, R value increases from 1.9048 for the neutral case of vi3-is-25 to 1.9145 for the emergency (vi3-is-249) and to 1.9462 for the mayday (vi3-is-440) cases. The decrease in R for vi3-is-415 and vi3-is-450 may have arisen because of the short duration of these two utterances. With a 601-sample bandpass filter, an utterance of fewer than 3000 samples has only

2400 samples in the steady portion of the filtered signal. As a result, energies of the filtered signal and the envelope may both be reduced, with the envelope having greater reduction. To assess the usefulness of R, therefore, more data with longer or of the same duration may be needed. Flatness ratio f_a, on the other hand, appears to decrease with stress. In the absence of a quantifying measure for stress, such as heart rate or blood pressure, however, f_a can only give a relative measure of stress.

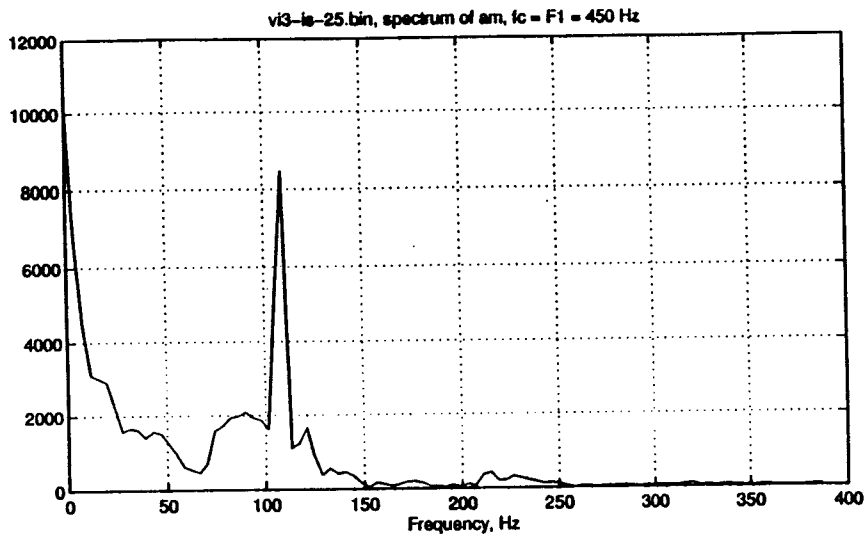
The degree of fluctuations in the AE and the IF is better reflected in the spectrum of these signals. The first 10 peak frequencies in the spectrum of the AE were consistently higher for the utterance in vi3-is-415 (when the pilot first realized the emergency and was losing altitude) than for all the other utterances. However, the spectral peak frequencies for other utterances did not show any distinct change relative to stress. The strongest spectral peak of the envelope occurred around 10 Hz for all cases except for vi3-is-415 for which it was approximately 30 Hz. Because of the muscle tremor frequency of approximately 10 Hz for unstressed speech, it appears that the peak envelope frequency may be related to the tremor frequency. More studies are needed to verify this aspect. The spectrum of the instantaneous frequency, on the other hand, appeared to have no correlation with stress.

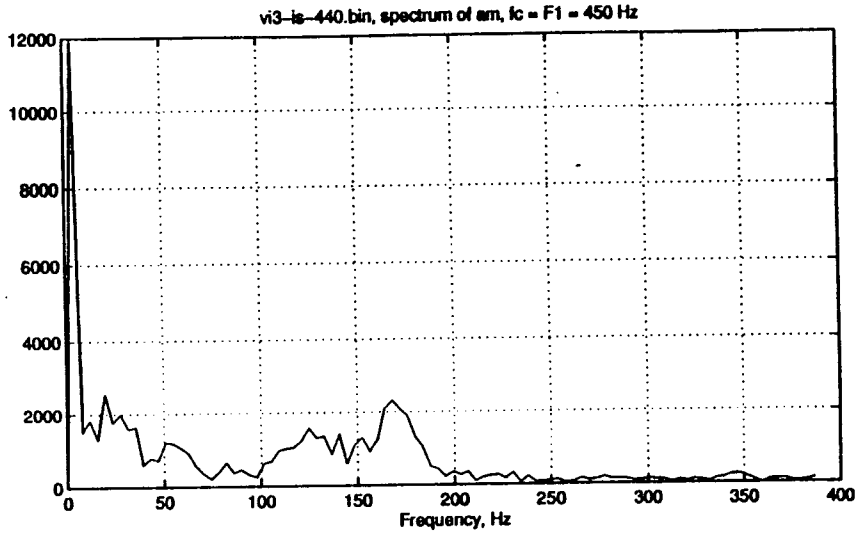
Demodulation in the vicinity of F1: The nominal first formant for the 'unstressed' utterance, vi3-is-25 was found to be around 450 Hz. Table III shows the energy ratio R and the flatness ratio f_a for the pilot's utterances using demodulation around 450 Hz. The following general observations were made from the demodulation characteristics. The maximum amplitude and the local variation of the envelope decreased with increasing stress. The local variation of the instantaneous frequency also decreased with increasing stress. More significantly, the two ratios also showed a decreasing trend with increasing stress; the R ratio for vi3-is-450, when the pilot declared "I'm engine out," however, showed a slight increase over the 'unstressed' case. It may be inaccurate to interpret from the low flatness measure for vi3-is-450 that the pilot was under more stress than he was while declaring mayday, or first realizing the emergency. The smaller values of f_a at higher stress levels show that the amplitude envelopes have higher fluctuations. This feature appears to contradict the general observation that the muscle tremor of about 10 Hz disappears when a person is under stress.

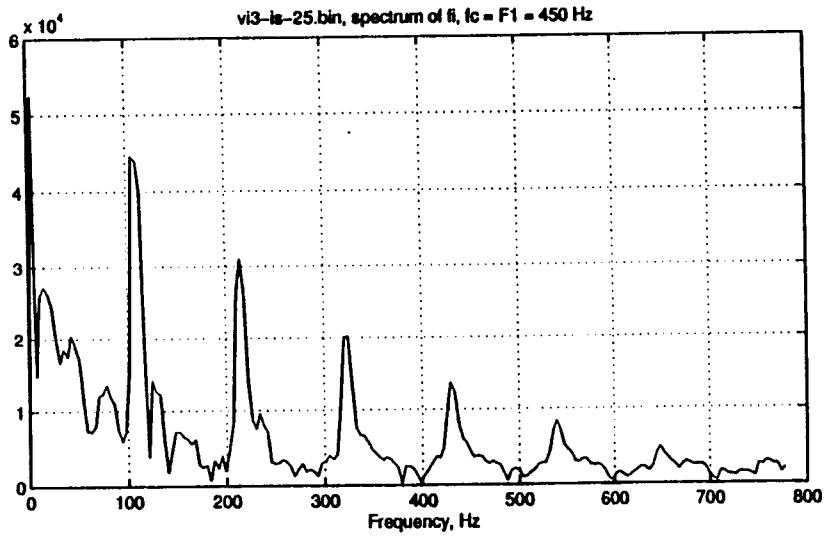
Table III

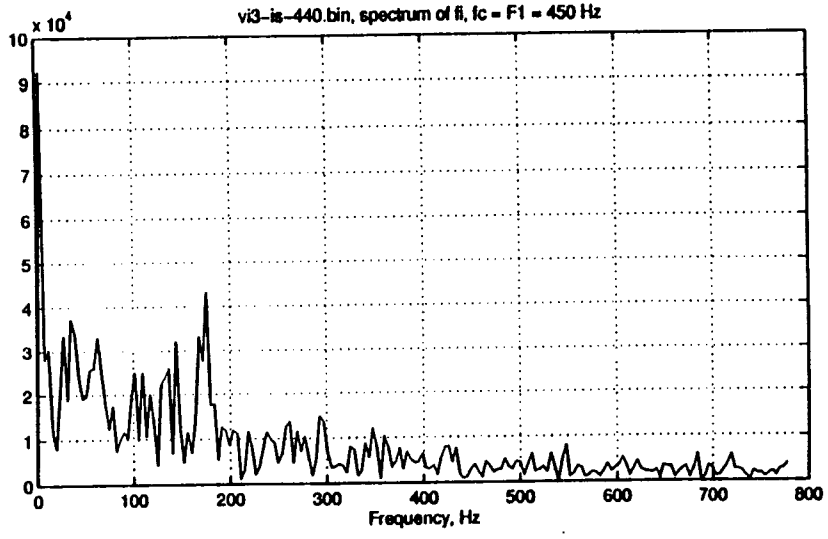
R and f_a values for Viper3 around $f_c = F1 \approx 450$ Hz

File name	R	f_a
vi3-is-25	1.9606	0.4901
vi3-is-249	1.9627	0.3534
vi3-is-415	1.9257	0.2614
vi3-is-440	1.9378	0.2373
vi3-is-450	1.9878	0.1793









(a) (b)

Fig. 2 Spectra of AE and IF around $f_c = 450$ Hz $\approx$ F1 for (a) vi3-is-25, and (b) vi3-is-440

At 'low' stress levels, the spectra of both the AE and the IF showed distinct harmonics approximating the fundamental frequency. This agrees with the observation made by Maragos, et al [11] that the AM envelope varies approximately with the fundamental frequency. As the stress level increased, however, the spectra showed less distinct peaks, which made it difficult to determine the fundamental frequency. Fig. 2 shows the spectra of AE and IF for vi3-is-25 (low stress) and vi3-is-440 (high stress). Fig. 2a clearly shows the fundamental frequency at approximately 115 Hz in the spectra of both AE and IF. From the second major peaks in the spectra in Fig. 2b, the pitch appears to be at approximately 170 Hz. The short duration of the utterance (about 220 ms for vi3-is-440 compared to about 500 ms for vi3-is-25) accounts for fewer pitch intervals in the spectra of AE and IF signals for vi3-is-440. However, the lack of clarity in resolving F0 as stress level increased, was seen in the other files as well. On the other hand, an elevated fundamental frequency is in itself a good indicator of high stress level.

Demodulation in the vicinity of F2: It has been observed [13] that the vowel spectrum is modified at high frequencies in stressful situations. This modification alters the resonant (formant) frequencies as well as the energy in higher frequency bands. Because of these changes at high frequencies with stress, modulation characteristics at 1400 Hz were studied. The frequency of f_c = 1400 Hz was chosen as an approximation to the second formant for the pilot.

The energy ratio R and the flatness ratio f_a, given in Table IV, do not appear to bear any relationship to perceived stress. The peak frequencies in the spectra of the AE, however, increased with stress. In addition, the largest peak in the spectra approximately tracked the stress-induced variation in the fundamental frequency, as shown in Table V. While the peak spectral frequency of the AE does not appear to track with F0 for the last two cases, the spectra for these cases do show significantly large peaks at frequencies of 175 Hz and 152 Hz that are closer to the range of fundamental frequencies.

Table IV

R and f_a values for Viper3 around f_c = F2 ≈ 1400 Hz

File name	R	f_a
vi3-is-25	1.9808	0.6992
vi3-is-249	1.9923	0.7016
vi3-is-415	1.9778	0.6325
vi3-is-440	1.9877	0.6964
vi3-is-450	1.9919	0.5761

Table V

Frequencies of the largest peak in the magnitude spectra of AE and the range of fundamental frequencies for Viper3 around $f_c = F2 \approx 1400$ Hz

File name	Freq. of largest spectral peak of AE, Hz	Range of Fundamental Frequency, Hz
vi3-is-25	125	101-118
vi3-is-249	98	89-92
vi3-is-415	113	111-125
vi3-is-440	86	114-173
vi3-is-450	59	151-157

Other Speaker and Features Studied: In addition to pilot Viper3 of the aircraft, the utterances of his wingman, Viper 4 were also analyzed. These utterances were extracted to include low stress and high stress situations based on the exchange between the two pilots. Although the wingman was clearly not at the same levels of stress, demodulation characteristics around $f_c =$ 1300 Hz (in the vicinity of the second formant) showed similar variations of AE, IF, R and f_a with stress as with Viper3.

Linear prediction (LP) model of the amplitude envelope: Based on the observation that pitch is a predominant stress indicator, it was hypothesized that the variation in the fundamental frequency of a speaker, or the pitch jitter, is due to a secondary linear system model. An all-pole linear prediction model of the amplitude envelope was considered to represent the secondary linear system - similar to the vocal tract - that produced the muscle tremor. A 31^{st} order linear prediction model was fit to the AE (Fig. 3) in an effort to estimate the stress-dependent muscle tremor frequency. It was conjectured that the spectral peaks in the system model would vary in accordance with stress, analogous to the resonant frequencies and shape of the vocal tract with different phonemes. With the lack of any spectral peaks in the vicinity of the tremor frequency of 10 Hz, however, the model did not support this conjecture. The spectrum did distinguish between stressed and unstressed utterances. For the utterances with no or low stress, i.e., vi3-is-25 and vi3-is-249, there were barely discernible, blended peaks at approximately 99 Hz and 117 Hz, respectively, when demodulated at the center frequency of $f_c = 450$ Hz. Although no peak was noticed around the fundamental frequency for the high stress case of vi3-is-415 at $f_c = 450$ Hz, a peak around 99 Hz was visible when the demodulation was carried out at $f_c = 3300$ Hz. For the utterances of vi3-is-440 and vi3-is-450, peaks around 164 Hz and 130 Hz, respectively, were detected with $f_c = 450$ Hz as well $f_c = 3300$ Hz. We notice that these frequencies are approximately the same as the fundamental frequencies in most cases. Thus it appears that the linear prediction model of the AE may be better able to follow the stress-dependent pitch variation. We also point out that in addition to the (blended/barely detectable) peak around F0, there was, in all cases, a peak around twice the center frequency f_c. The location of this peak at such a high frequency cannot be accounted for, except for being an artifact of the nonlinear

demodulation process. The ability to resolve the blended peak around F0 using a higher center frequency f_c for demodulation is worth pursuing further.

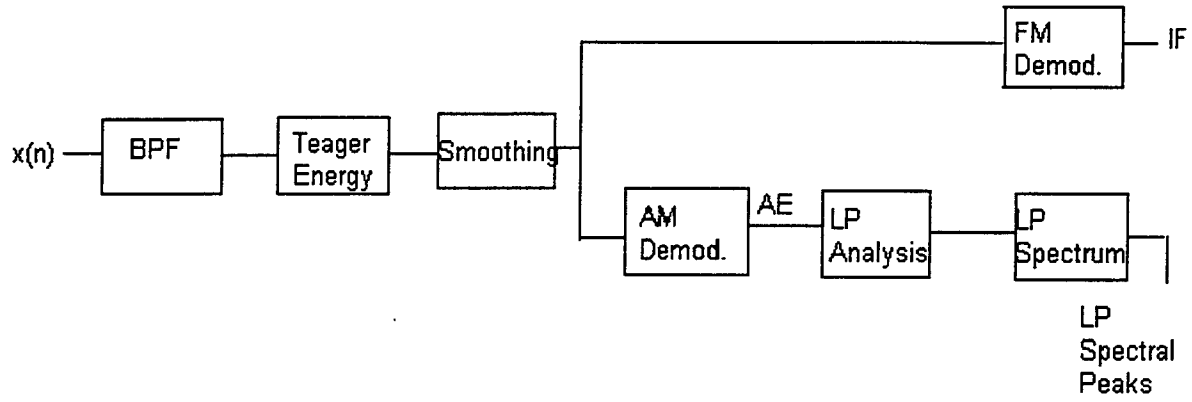


Fig. 3 Process of finding LP spectrum of AE

Features used in a stress classification process using a neural network: The following features derived from the demodulation characteristics at f_c = 130 Hz and also at 3400 Hz were used to classify stress in the SUSAS database. (1) First five peaks in the IF, (2) AE values at these peak frequencies, (3) First five peaks in the AE, (4) IF values at these peak amplitudes, (5) First five peak frequencies in the spectrum of AE, and (6) the energy ratio, R. Using the 26-element feature vectors from three utterances as reference for each of 'low' (with a stress level assignment of 1), 'medium' (stress level = 2), and 'high' (stress level = 3) utterances by a speaker speaking the same word, a two layer neural network was simulated in Matlab. Although the validation process worked well, test utterances from the same speaker yielded only two correct classification out of seven. A different choice of features with more reference vectors with a better network or classifier may yield higher stress classification score. In addition, it must be pointed out that the labeling of stress as low or high based on utterances during a roller-coaster ride may not be indicative of actual stress. Therefore, studies on more utterances from speakers under psychological stress are needed for definitive testing of stress classification.

Discussion and Further Work

Demodulation of bandpass-filtered speech in the vicinity of a resonant frequency of vocal tract has been shown to bear stress-related information. To quantify by getting a relative measure of the stress from the acoustic features, an independent measure such as the heart rate or blood pressure of the speaker during the utterances at various levels of stress is needed. A European database with heart rates of fighter aircraft controllers for some utterances has been located. Although it is incomplete in physical stress measures, modulation analysis of this database may help determine the efficacy of the AM-FM features in stress detection. In addition, the following tasks are suggested for further work in the study of modulation-based stressed speech analysis.

a. Analysis of pitch and formant variation from the AE and IF around chosen Frequencies: Inasmuch as pitch and formant are known to vary with stress, this study may help assess stress without first determining F0 and formants. The analysis may determine the demodulation characteristics at frequencies in a broad neighborhood of F0 or any formant and narrow the range based on the AE and IF behavior.
b. 'Lie detection': Detection of deception by law enforcement personnel using polygraph relies on various physical measurements such as heart rate and blood pressure. If deception increases a speaker's stress level, the increase may manifest in the modulation features. Clearly, employing speech feature extraction is preferable as a non intrusive tool for 'lie detection.' Therefore, using a database of true and deceitful statements, the modulation features may be analyzed to correlate with truth and deception.
c. Better methods of demodulating at a given center frequency, especially around F0: Methods using Hilbert transform [14] or Wigner distribution [15] may be studied for analyzing stressed speech with error-free determination of AE and IF. This study is particularly warranted at low center frequencies (i.e., around F0) because of the high error introduced in the IF by the algorithm currently employed [15].
d. Robust features for speaker identification: The spectral peaks in the AE are seen to be independent of stress level, varying only with the center frequency. This aspect needs further study before employing the peaks as features for speaker identification.

Other related areas of study may include pitch estimation from the AE and its linear prediction model or IF, demodulation analysis at higher formants available, demodulation at frequencies other than the fundamental frequency and formants of stressed and unstressed speech, and demodulation of fricatives and nasals.

Conclusion

An analysis of stressed speech using AM and FM demodulation characteristics was carried out. From the preliminary results using a single speaker's utterances under actual stress from an aviation emergency, the demodulation characteristics show promise as a viable tool for detecting stress. Further studies with many speakers and/or many stressed speech utterances are needed to determine the efficacy of the modulation-based features in quantifying relative levels of stress.

References

[1] C.E. Williams, and K.N. Stevens, "Vocal Correlates of Emotional Stress," in Speech Evaluation Psychiatry, J.K Darby, Jr. (Ed.), Grune & Stratton, Inc.,1981.

[2] E. Absil, et al., "Time-related Variabilities in Stressed Speech under Laboratory and Real Conditions," Technical Proceedings, Workshop on Speech under Stress Conditions, NATO Defence Research Group, pp. 23-1 - 23-4.

[3] P. Benson, "Analysis of the Acoustic Correlates of Stress from an Operational Aviation Emergency," *ibid*, pp. 25-1 - 25-4.

[4] A. Protopapas, and P. Lieberman, "Effects of Vocal F0 Manipulations on Perceived Emotional Stress," *ibid*, pp. 4-1 - 4-4.

[5] J-C Junqua, "The Influence of Acouistics on Speech Production: A Noise-Induced Stress Phenomenon known as the Lombard Reflex," *ibid*, pp. 14-1 - 14-8.

[6] S. Arnfield, , et al, "Emotional Stress and Speech Tempo Variation," *ibid*, pp. 7-1 - 7-3.

[7] Teager, H.M., and S. Teager, "Evidence for Nonlinear Production Mechanisms in the Vocal Tract," NATO Advanced Study Inst. On Speech Production and Speech Modeling, Bonas, France, 1989, Kluwer Acad. Pub., 1990.

[8] G.Zhou, J.H.L. Hansen and J.F. Kaiser, "Classification of Speech under Stress based on Features derived from the Nonlinear Teager Energy Operator," Proc. IEEE ICASSP '98, pp. 549-552, 1998.

[9] A.B. Fineberg, R.J. Mammone and J.L. Flanagan, "Application of the Modulation Model to Speech Recognition," Proc. ICASSP '92, pp. I-541-I-544, 1992.

[10] J.F. Kaiser, "On a Simple Algorithm to Calculate the 'Energy' of a Signal, Proc. ICASSP '90, pp. 381-384, 1990.

[11] P. Maragos, T.F. Quatieri, and J.F. Kaiser, "Speech Nonlinearities, Modulations, and Energy Operators," Proc. ICASSP '91, pp. 421-424, 1991.

[12] P. Maragos, J.F. Kaiser and T.F. Quatieri, "On Separating Amplitude from Frequency Modulations using Energy Operators," Proc. IEEE ICASSP '92, Vol.2, pp. 1-4, 1992.

[13] R. Ruiz and C. Legros, "Vowel Spectral Characteristics to Detect Vocal Stress," Proc. 15th Int. Congress on Acoustics, Trondheim, Norway, pp. 141-144, June 1995.

[14] D. Vakman, "On the Analytic Signal, the Teager-Kaiser Energy Algorithm, and Other methods for Defining Amplitude and Frequency," IEEE Trans. Signal Processing, Vol. 44, No. 4, pp. 791-797, Apr. 1996.

[15] P. Rao, "A Robust Method for the Estimation of Formant Frequency Modulation Speech Signals," Proc. IEEE ICASSP '96, pp. 813-816, 1996.

A STUDY ON ACCELERATING THE RAY/TRIANGULAR-FACET INTERSECTION COMPUTATION IN XPATCH

Donald L. Hung
Assistant Professor
School of Electrical Engineering and Computer Science

Washington State University, Tri-Cities
2710 University Drive
Richland, WA 99352

Final Report for:
Summer Faculty Research Program
Information Directorate (IFSD)
Wright Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Wright Research Site

August 1998

A STUDY ON ACCELERATING THE RAY/TRIANGULAR-FACET INTERSECTION COMPUTATION IN XPATCH

Donald L. Hung
Assistant Professor
School of Electrical Engineering and Computer Science
Washington State University, Tri-Cities

Abstract

Xpatch is a software program that uses the ray-tracing technique to model the incident radar signal and generates radar cross section predictions of various objects. The object surfaces are modeled by shape primitives such as boxes, triangular facets, bicubic bezier patches, etc.. The heart of the ray-tracing program is to determine if a given ray intersects with some given shape primitives. Despite of its power, the ray-tracing technique suffers from its computational complexity and intensity. This study investigates the possibilities of accelerating the ray/triangular-facet intersection computation, by adopting effective algorithms and developing specially tailored hardware for the execution of the algorithms. Estimated performance and feasibility are discussed. Suggestions on future work are also included.

A STUDY ON ACCELERATING THE RAY/TRIANGULAR-FACET INTERSECTION COMPUTATION IN XPATCH

Donald L. Hung

Introduction

A significant amount of work has been conducted at the Air Force Research Lab (AFRL) to determine what part of the Xpatch program is the most time consuming. The finding is, the majority of the time (between 50% and 80%) is spent in executing the ray-tracing algorithms [1]. One of the major drawbacks of the ray-tracing technique is the exorbitant computation required. Unfortunately, despite of many efforts which have been made in the past [2-7], and the computing power of the recent computers, execution of ray-tracing algorithms remains as a computational bottleneck. The focus of this study is on seeking hardware solutions for accelerating the ray/triangular-facet intersection computation. Two approaches have been taken. The first one looks into the algorithms. The goal is to find the ones that can be executed effectively by hardware. The second approach explores architectural alternatives for hardware implementation. The goal is to maximize the throughput by fully utilizing the parallelism in the algorithms and streamlining the execution.

The computation for ray/triangular-facet intersection needs to determine if a given ray hits a given triangular facet. It includes two parts: 1) the computation for finding the intersection point of a given ray and a plane determined by a given triangular facet, 2) the computation for determining if the ray/plane intersection point is inside the given triangular facet. After some careful study, it was concluded that it would be difficult to find a replacement for the ray/plane intersection algorithm currently used by the Xpatch. On the other hand, it was found that there are different ways to determine if a ray/plane intersection point falls inside a triangle on that plane. Based on Haines' ray/polygon intersection algorithm [8], a ray/triangle intersection algorithm was proposed and some special hardware functional units needed by the execution of the algorithm were designed. The algorithms for ray/plane and ray/triangle intersections, and the hardware functional units developed for the proposed algorithm are described in sections Algorithms and Special Functional Units, respectively.

For architectural exploration of the custom hardware, without specifications on data format and software/hardware interfacing scheme, an initial unconstrained scheduling was conducted which reveals the inherent parallelisms in the algorithms. Based on this and a hypothetical software/hardware interfacing scheme, a second version of the scheduling was proposed. In which the operations were streamlined and therefore could be implemented with fine-grained pipelining. This part of the work, together with estimations on performance and costs of the hardware, are included in the section Architectural Exploration.

The last section, Conclusions and Future Work, presents the conclusion of this study and recommends future research work.

Algorithms

The computation for ray/triangular-facet intersection includes two algorithms: 1) the ray/plane intersection algorithm, and 2) the ray/triangle intersection algorithm.

The Ray/Plane Intersection Algorithm

The purpose of this algorithm is to find the intersection point of a given ray and a plane that is determined by a given triangle. For generality, both the ray and the plane are in a 3-D space.

The plane is described by the following equation:

$$Ax + By + Cz + D = 0 \tag{1}$$

The ray is described by the expression below:

$$R = R_0 + R_d t \tag{2}$$

where $R_0 = [x_0 \quad y_0 \quad z_0]$ is the origin of the ray and $R_d = [x_d \quad y_d \quad z_d]$ is the direction of the ray. With the distance parameter t, any point $[x \ y \ z]$ on the ray R can be presented as

$$x = x_0 + x_d t \tag{3}$$

$$y = y_0 + y_d t \tag{4}$$

$$z = z_0 + z_d t \tag{5}$$

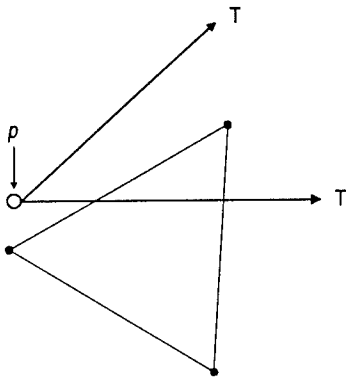
The ray/plane intersection point can be found by solving the plane and ray equations simultaneously with respect to the parameter t, and then using Equations (3) through (5) to determine the values of x, y, and z. Step-by-step executions of the algorithm is listed below:

1) Calculate $v_d = Ax_d + By_d + Cz_d$ and,
   if $v_d = 0$, there is no intersection (ray in parallel with the plane);
   if $v_d > 0$, the plane is pointing away from the ray;
   if $v_d < 0$, the ray and the plane have an intersection point.
2) Calculate $v_0 = -(Ax_0 + By_0 + Cz_0 + D)$.
3) Calculate the value of t at the point of intersection: $t^* = v_0 / v_d$ and,
   if $t^* < 0$, there is no intersection (plane behind the ray);
   if $t^* = 0$, ray originated from the plane;
   if $t^* > 0$, the ray and the plane have an intersection point.
4) Calculate the ray/plane intersection point $R_I = [x_I \quad y_I \quad z_I]$ where,
   $x_I = x_0 + x_d t^*$, $y_I = y_0 + y_d t^*$, $z_I = z_0 + z_d t^*$.

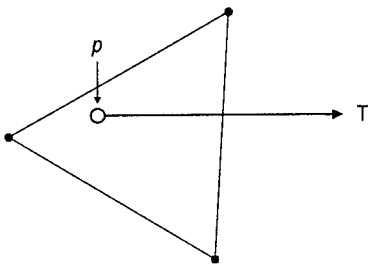
It is clear that the ray/plane intersection algorithm requires a substantial amount of arithmetic operations.

The Ray/Triangle Intersection Algorithm

The purpose of this algorithm is to determine if R_I, the ray/plane intersection point obtained from the previous algorithm, is inside a given triangle on that plane. The algorithm presented here is based on Haines' ray/polygon intersection algorithm. The idea is to first project the ray and the triangular facet onto a plane that is orthogonal to the dominant coordinate in the plane equation (The dominant coordinate is the one corresponding to A, B, or C in Eqn. (1) whichever has the greatest magnitude.). Since this is a topology-preserving projection, we can then apply Jordan's curve theory in the 2-D space to determine if the ray/plane intersection point R_I is inside the triangular facet. Jordan's curve theory is illustrated by Fig. 1 shown below:



(a) The testing vector T crosses the edges of the triangle twice, or does not cross at all, the point p is outside of the triangle.

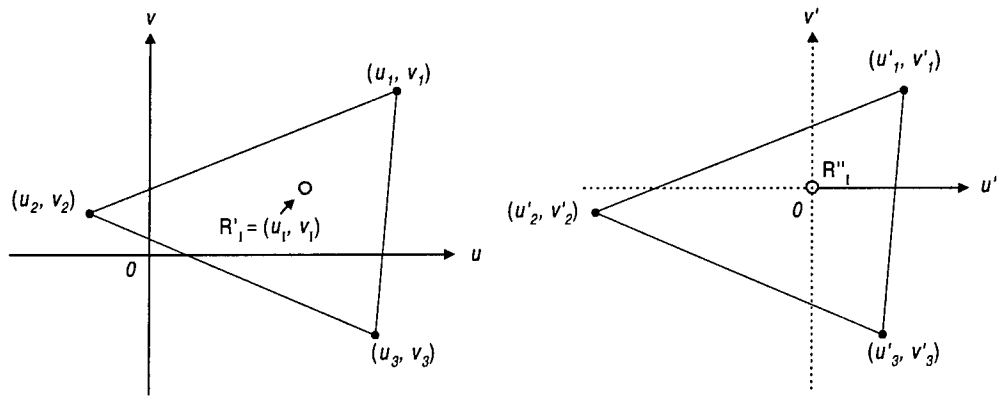


(b) The testing vector T crosses the edges of the triangle once, the point p is inside of the triangle.

Fig. 1. A graphic illustration on Jordan's curve theory.

In order to apply Jordan's curve theory to our case, once the ray/plane intersection point R_I and the triangular facet are projected to a 2-D space with coordinates (u,v), we need to shift the origin of the 2-D space to where the intersection point R_I is located. This is illustrated by Fig. 2 shown on the next page. In Fig. 2(a), the point $R'_I = (u_I, v_I)$ is projected from the original ray/plane intersection point R_I in a 3-D space, and the triangle defined by the three vertices (u_1, v_1), (u_2, v_2) and (u_3, v_3) corresponds to the original 3-D triangular facet. Fig. 2(b) shows that after the origin of the 2-D coordinate being shifted to the projected ray/plane intersection point, we are ready to use the Jordan's curve theory. Since the origin (the ray/plane intersection point) in Fig. 2(b) can be viewed as the point p in Fig. 1, and the positive half of the axis u' in Fig. 2(b) can be viewed as the testing vector T in Fig. 1, respectively.

Based on the coordinate system shown in Fig. 2(b), a set of rules can be derived to determine if the ray/plane intersection point is inside a triangle. Given R_I, the original ray/plane intersection point in a 3-D space, and the three vertices T_1, T_2, and T_3, of the original triangular facet (also in the 3-D space), step-by-step executions of the ray/triangle intersection algorithm is listed below:



(a) Triangle and the ray/plane intersection point R_1 projected to a 2-D plane

(b) Origin of the 2-D plane has been shifted to the intersection point

Fig. 2. Using Jordan's curve theory to determine if a ray hits a triangular facet.

1) Determine the dominant coordinate which is correspondent to the one with the greatest magnitude among A, B, or C in the plane equation.

2) Project R_1, T_1, T_2, and T_3 to the 2-D plane. That is,

$R_1 \rightarrow R'_1$: $[x_1 \quad y_1 \quad z_1] \rightarrow [u_1 \quad v_1]$,

$T_1 \rightarrow T'_1$: $[x_1 \quad y_1 \quad z_1] \rightarrow [u_1 \quad v_1]$,

$T_2 \rightarrow T'_2$: $[x_2 \quad y_2 \quad z_2] \rightarrow [u_2 \quad v_2]$,

$T_3 \rightarrow T'_3$: $[x_3 \quad y_3 \quad z_3] \rightarrow [u_3 \quad v_3]$,

by dropping the coordinate corresponds to the dominant coordinate which is determined in Step (1).

3) Shift the origin of the 2-D coordinate to R'_1. That is, execute

$T'_i \rightarrow T''_i = [u'_i \quad v'_i] = [u_i - u_1 \quad v_i - v_1]$

for $i = 1, 2, 3$.

4) Calculate $\Delta u_i = u_i - u_j = u'_i - u'_j$ and $\Delta v_i = v'_i - v'_j = v_i - v_j$,

where $i = 1, 2, 3$ and $j = (i \bmod 3) + 1$.

5) For each edge of the 2-D triangle determined by a pair of vertices $[u'_i \quad v'_i]$ and $[u'_j \quad v'_j]$ (here $i = 1, 2, 3$ and $j = 1 + i \bmod 3$) obtained in Step (3), execute the following:

- find $cross_i = [/\text{sign}(u'_i) \cdot /\text{sign}(u'_j)][\text{sign}(v'_i) \oplus \text{sign}(v'_j)]$

  and $check\_more_i = [\text{sign}(u'_i) \oplus \text{sign}(u'_j)][\text{sign}(v'_i) \oplus \text{sign}(v'_j)]$,

- if $check\_more_i = 1$ then calculate $\delta_i = u'_i \Delta v_i - v'_i \Delta u_i$ and replace the value of $cross_i$ with

  $cross_i = \text{sign}(\Delta v_i)$ XNOR $\text{sign}(\delta_i)$.

6) Obtain the decision value HIT ($HIT = 1$ means the ray/plane intersection point is inside the triangular facet):

$HIT = cross_1 \oplus cross_2 \oplus cross_3$.

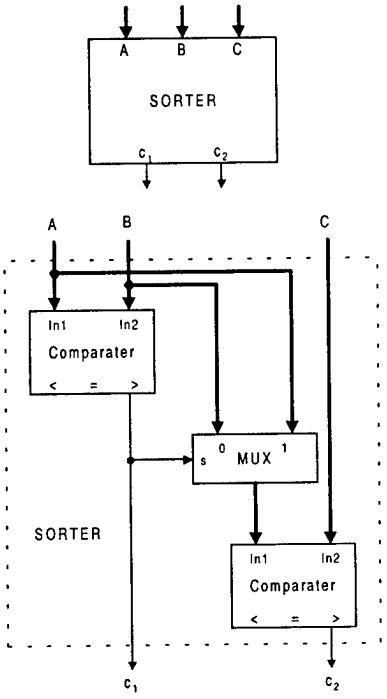
The above algorithm looks complicated, but it can be executed efficiently by hardware because: 1) most of the operations are non-arithmetic and suitable for digital hardware realization, and 2) many of the operations are on the sign bits only. Also note that the algorithm remains valid if one of the vertex is on the u' axis. The next section shows some specially designed hardware functional units for executing the proposed ray/triangle intersection algorithm.

Special Functional Units

In order to execute the proposed ray/triangle intersection algorithm with hardware, some special functional units are designed. They are described in this section.

Sorter

The purpose of this functional unit is to execute Step (1) of the proposed ray/triangle intersection algorithm. It takes the magnitudes of *A, B, C* from the plane equation as inputs, and generates a 2-bit control signal $c[2:1]$ that indicates which input corresponds to the dominant coordinate. Schematic and functionality of the *Sorter* is given in Fig. 3.

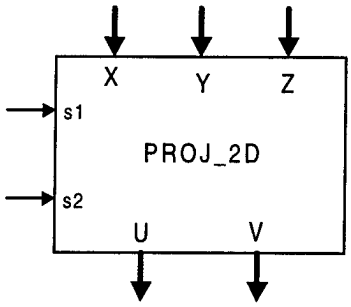


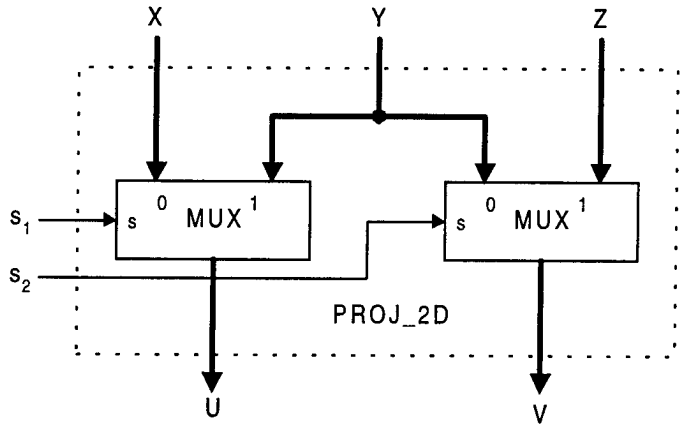
c_1	c_2	Sorted Order	The MAX Is
0	0	(A < B) < C	C
0	1	(A < B) > C	B
1	0	(A > B) < C	C
1	1	(A > B) > C	A

Fig. 3. Functional unit Sorter.

Proj_2d

The functional unit *Proj_2d* was designed to work with the *Sorter* to execute Step (2) of the proposed ray/triangle intersection algorithm. Controlled by the *Sorter*'s output signal $c[2:1]$, the operation of 3-D to 2-D projection can be accomplished with *Proj_2d.* Schematic and functionality of the unit is shown in Fig. 4. Signal mappings between the *Proj_2d* and the *Sorter* are: $s_1 = c_1 \cdot c_2$ and $s_2 = c_2$.





s_1	s_2	Passed 2-D Data
0	0	X, Y
0	1	X, Z
1	0	Prohibited
1	1	Y, Z

Fig. 4. Functional unit Proj_2d.

Q_check

The functional unit *Q_check* executes part of the Step (5) of the proposed ray/triangle intersection algorithm. It takes the sign bits from a pair of the vertices (of the 2-D triangle) and generates the corresponding $cross_i$ and $check\_more_i$ signals. Schematic of the *Q_check* is given in Fig. 5.

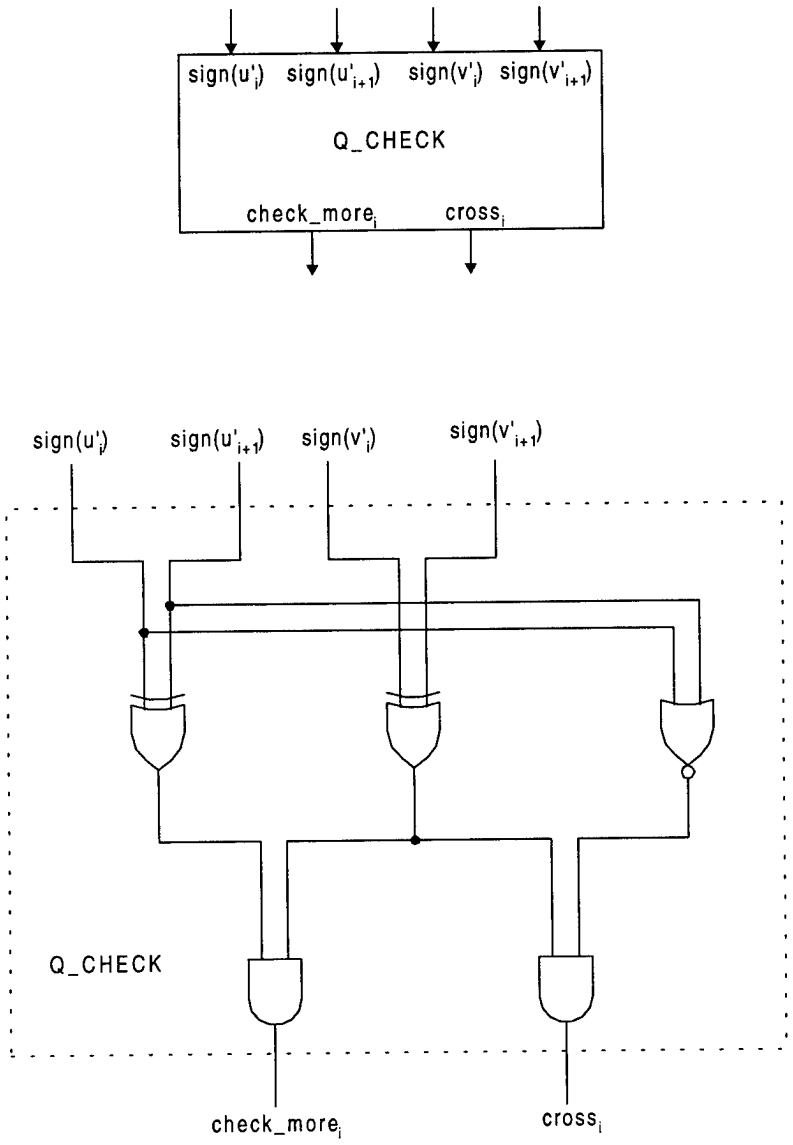


Fig. 5. Functional unit Q_check.

HIT

The functional unit *HIT* executes Step (6) and part of the Step (5) of the proposed ray/triangle intersection algorithm. Its schematic is shown in Fig. 6.

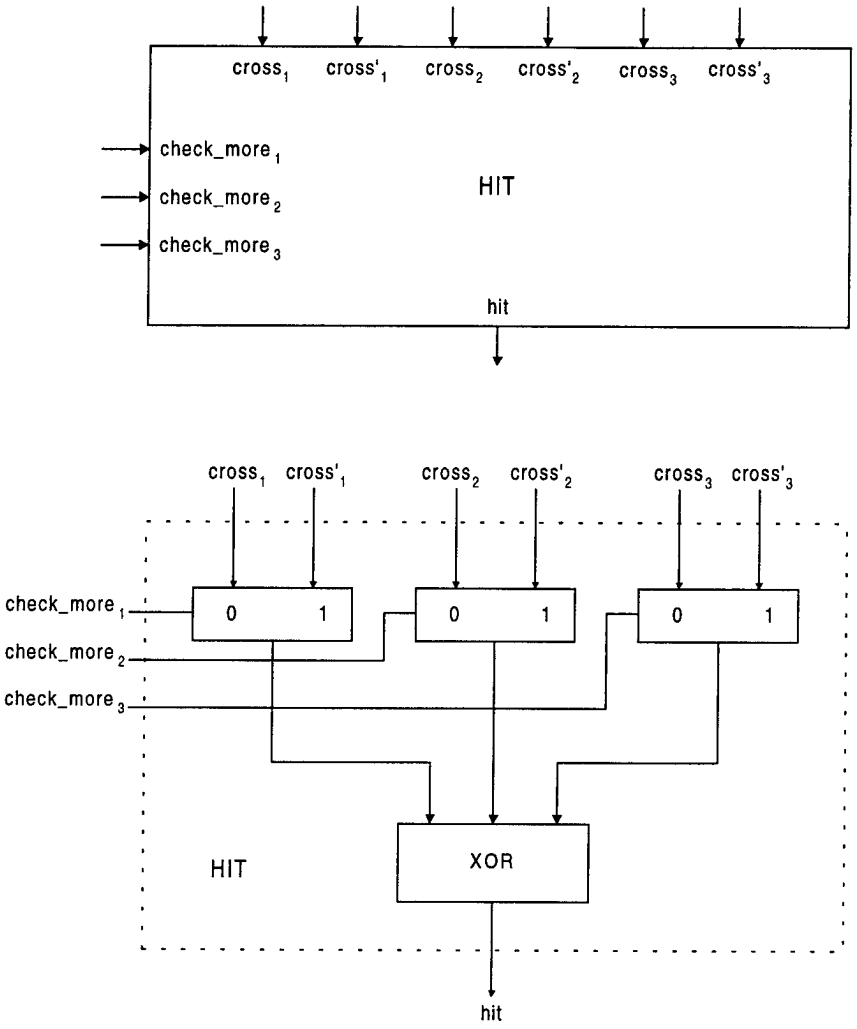


Fig. 6. Functional unit HIT.

Architectural Exploration

As a preparation for hardware design and implementation, architectural exploration was conducted at the scheduling level. Two versions of the scheduling are included here. The first version is shown in Fig. 7. It does not take any resource or timing constraints into consideration, and assumes that all data (ray, plane and triangle) needed for computation are available at the beginning. The ASAP scheduling approach was adopted. The initial schedule in Fig. 7 reveals all possible parallelisms in the algorithms. It also shows that the computation (and therefore the hardware) requirement is dominated by the ray/plane algorithm. If the hardware accelerator needs to deal with discrete inputs (for a single triangular facet) only, the computation and hardware overhead for the proposed ray/triangle algorithm will be very limited. The reason is, all special functional units are cheap and fast and, when arithmetic functional units (subtractors and multipliers) are needed, they can be shared with the ray/plane algorithm.

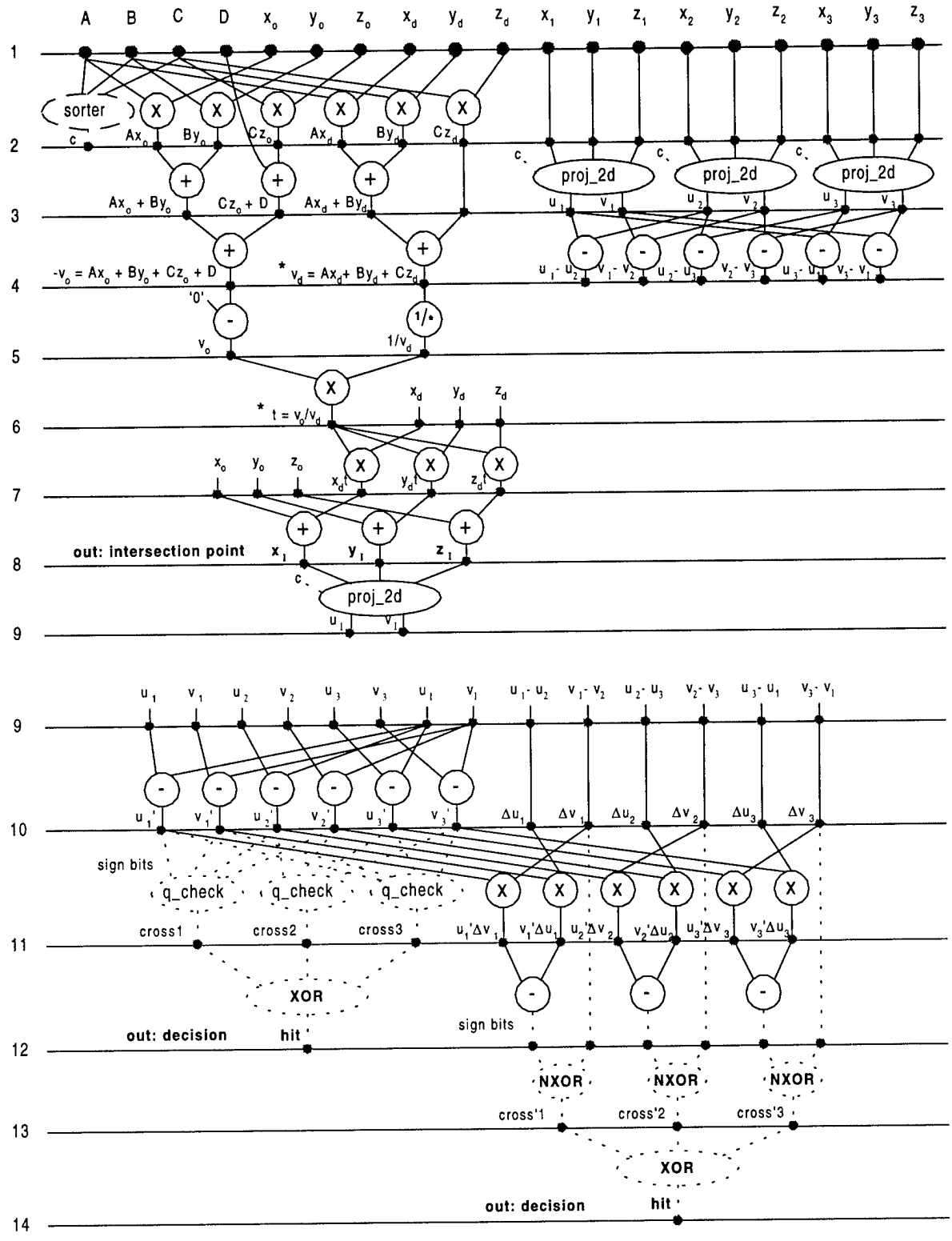


Fig 7. Unconstrained ASAP scheduling for the ray/plane/triangle intersection algorithms

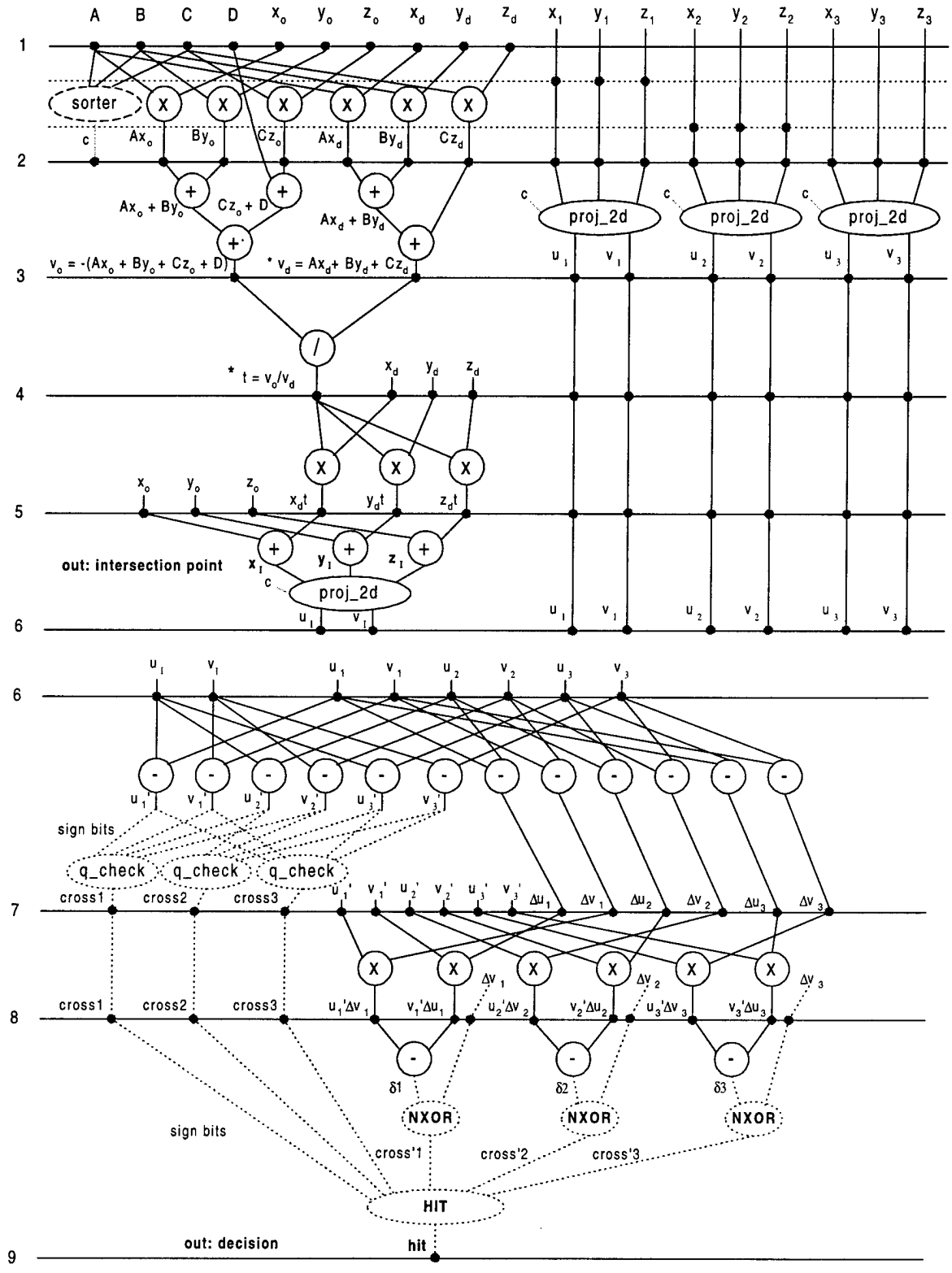


Fig. 8. The second scheduling for pipelined implementation of the ray/plane/triangle algorithms.

Based on the initial operation schedule shown in Fig. 7, a second scheduling was proposed and its corresponding DFG is shown in Fig. 8. As specifications for data format and software/hardware interface were not available, the scheduling was performed based on the following assumptions:

- The hardware accelerator needs to deal with continuous input data streams (e.g., a testing ray against multiple triangular facets).
- High throughput is the primary concern.
- There is no hardware resource constraint.

It was then decided that multiple input ports must be used. The reason is, excluding the data associated with the given ray (x_0, y_0, z_0, x_d, y_d, z_d), there are 13 plane/triangular facet data (A, B, C, D, x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3) that need to be brought in for computation, and therefore the overall throughput will be constrained by the data input rate. It is clear that in order to keep up with the high input data rate, a pipelined approach must be adopted. The scheduling DFG shown in Fig. 8 assumes 4 input ports. During the first stage of the DFG, the system uses 4 cycles to read in a set of plane/triangular-facet data (with respect to a given ray). Note that this does not mean that the rest of the stages all have 4 cycles. The stages shown in Fig. 8 were divided for ease of understanding. In fact each of those stages may include different numbers of cycles. The granularity of the pipeline will be determined by performance specification and the nature of the hardware functional units, which in turn depend on the data format.

For best performance, no busing and resource sharing are considered. Inter-stage communication will be through point-to-point interconnection. The major hardware cost will come from inter-stage registers and the following arithmetic functional units:

- Multiplier 15
- Divider 1
- Adder/subtractor 23

Assuming using 32-bit data, the estimated hardware complexity is at 300,000- to 400,000-gate level. Power consumption of the system will be high due to pipelining.

Assuming ultra high-speed data input rate would be available at 100 MHz, with 4 input ports, the maximum achievable system throughput is 25 x 10^6 per second. As a comparison, throughput on a 200 MHz Sun Ultra SPARK processor is around 4 x 10^5 per second [1].

Conclusion and Future Work

Ray-tracing is a power technique but its wide-spread application could be hampered without acceleration. This study explores the possibility of accelerating the ray/triangular-facet computation using custom hardware that

directly supports carefully selected algorithms. The finding is, this approach could deliver significant speedups (greater then 60) over the high-end general-purpose computers, based on the current semiconductor technology. To carry on this work, the following tasks are recommended:

- Test the proposed ray/triangle intersection algorithm in software,
- Streamline the overall software program flow to define the software/hardware boundary and interfacing scheme,
- Find the dynamic range of the run-time data and define the data format (urgent),
- Specify performance requirement based on application needs,
- Design (or buy) and test pipelinable arithmetic functional units,
- Design the control/steering circuits,
- Integrate and prototype the overall hardware system,
- Integrate hardware with software.

After all, effort for developing efficient algorithms is always crucial and highly recommended.

References

[1] Private communications with Brian A. Kadrovach, AFRL/IFSD, June 1998.

[2] B. Arnalldi, T. Priol and K. Bouatouch, "A new space subdivision method for ray tracing CSG modelled scenes," *The Visual Computer*, Springer-Verlag, vol. 3, pp.98-108, 1987.

[3] M.A.J. Sweely and R.H. Bartels, "Ray tracing free-from B-spline surfaces," *IEEE Comput. Graph. Appl.* 6(2), 41-49, Feb. 1986.

[4] J.G. Clearly, et.al., "Multiprocessor ray tracing," *Comput. Graph., For.* 5, 3-12, 1985.

[5] M. Dippe and J. Swensen, "An adaptive subdivision algorithm and parallel architecture for realistic image synthesis," *Comput. Graph.* 18(3), 149-158, July 1984.

[6] J. Goldsmith and J. Salmon, "A ray tracing system for the hypercube," *Caltech Concurrent Computing Project Memorandum HM154*, California Institute of Technology, 1985.

[7] H. Kobayashi, et.al., "Parallel processing of an object space for image synthesis using ray tracing," *The Visual Computer*, Springer-Verlag, vol. 3, No. 1, pp. 13-22, 1987.

[8] E. Haines, "Essential ray tracing algorithms," Chapter 2 in *An Introduction to Ray Tracing*, A.S. Glassner Ed., Academic Press, 1989.

Acknowledgment

This work was supported by the Air Force Office of Scientific Research, whose support is gratefully acknowledged. I would like to thank my team leader Luis Concha and teammates Brian Kadrovach, Britton Read, Keith Pedersen and David Bawcom at the AFRL/IFSD branch, for their support, collaboration, assistance and encouragement throughout the work. I also want to thank Darrell Barker, Bob Ewing, Frank Yong, Gregory Tumbush, Maya Rubeiz and Gary Fecher of the AFRL/IFTA branch, for their interests in this work, and/or the helpful meetings and discussions with me during the summer of 1998.

On a wavelet-based method of watermarking digital images

Adam Lutoborski
Professor of Mathematics
Department of Mathematics
Syracuse University
Syracuse, New York 13244-1150

Lt Arnold Baldoza and John Vergis
Air Force Research Laboratory/IFEC
32 Brooks Road, Rome, New York 13441-4114

Final Report for:
Summer Faculty Research Program
Air Force Research Laboratory
Rome Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC
and
Air Force Research Laboratory
Rome Research Site

July 1998

On a wavelet-based method of watermarking digital images

Adam Lutoborski
Department of Mathematics
Syracuse University

Lt Arnold Baldoza and John Vergis
Multi-Sensor Exploitation Branch
Air Force Research Laboratory

Abstract

We present two wavelet-based watermarking methods for digital images. The first method uses the singular value decomposition of the matrix of wavelet coefficients of the image to generate the watermark; the second method uses cellular automaton. Both methods provide a high degree of security and robustness to various image processing operations and both require the original image for detection of the watermark.

Introduction to digital watermarking

Digital watermarking is the embedding of unobtrusive markings into digital media (e.g., digital images, digital audio, and digital video). The embedded mark, the "watermark" which contains key information such as authentication or copyright codes, is generally invisible (or imperceptible). However, although the marked media is perceptually unchanged, the embedding method imposes modifications that are detectable by using an appropriate detection algorithm. Watermarks are embedded into a digital host in such a way that they withstand operations that do not degrade the host beyond its utility value. These operations include, but are not restricted to, common image processing operations such as compression, filtering, and blurring. Watermarks are embedded into a digital host in such a way that they are inseparable from it. Watermarking technology, if designed properly, can be used as proof of ownership (copyright protection), as a content authentication tool (tampering detection), and as a means of inserting digital fingerprints into the data to allow tracing unauthorized disclosures, see [WD], [CKLS], [SZT1, 2], [CL], [CMYY1, 2], [B].

Most of the recent work in watermarking can be can be grouped into two categories: spatial domain methods, and transform-based methods, [BGM].

If the pixels of an image are locally highly correlated, the idea of a useful transform is to look for a description of the image which uses a small number of parameters and yet which captures most of its independent features. We need a basis in which the coordinates of the image are almost uncorrelated (or very mildly correlated) and in which we need fewer coordinates to well represent an image. In most practical applications either the discrete Fourier transform (DFT) or the discrete wavelet transform (DWT) is computed resulting in the transform coefficients which in the case of DFT characterize the frequency content of the image and in the case of DWT the space-frequency content of the image, see [PBBC], [].

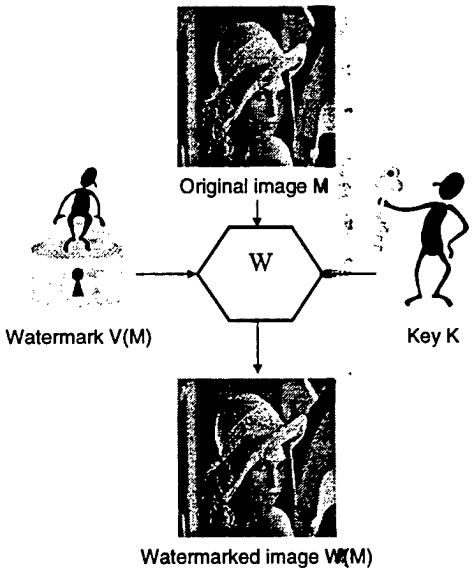


Figure 1. Generalized watermark insertion process.

Transform watermarking techniques in general result in a dense watermark affecting all pixels of the image despite the fact that the watermarking may be done only on a subset of the transform coefficients. One benefit of using the transform technique is that the watermark is robust with respect to all image operations which do not significantly affect the transform coefficients. For example, JPEG compression does not significantly affect the discrete cosine coefficients in the DFT. Moreover, wavelet compression preserves the largest wavelet coefficients. Another benefit is that the transform watermarking can be performed on the transformed (compressed) representation of the image.

Regardless of the category in which a particular watermarking technique falls into, watermarking methodologies follow a similar general embedding and detection scheme, [ZST].

To embed a watermark into an image (**Error! Reference source not found.**), consider digital images to be rectangular matrices with integer entries with values in the interval [0,255]. We denote the set of all digital images by $\boldsymbol{I}$. Given an image $M \in \boldsymbol{I}$, a digital watermark V is a sequence of bits (possibly an image) which when added to M produces the watermarked image $W(M)$, where $W(M) = M + V$. The algorithm which for a given image M constructs a watermarked image $W(M)$ defines the watermarking mapping $W: \boldsymbol{I} \rightarrow \boldsymbol{I}$

$$W: M \rightarrow W(M).$$

The digital watermark V is given by $V=V(M)=W(M)-M$. If the watermarking mapping W is invertible, then the original image M can be retrieved (without any loss) from the watermarked image $W(M)$.

We will say that the watermark $V(M)$ is detectable in N, if given the original image M, there exists a continuous detection function d: $\boldsymbol{I}\times\boldsymbol{I}\rightarrow[0,1]$ such that

$$d(M, N) = 1 \qquad \text{if } N = W(M)$$

$$0 \leq d(M, N) < 1 \text{ if } N \neq W(M)$$

for all $M, N \in \boldsymbol{I}$.

We do not address here the issue of detection of the watermark in the absence of the original image, see [CMYY2].

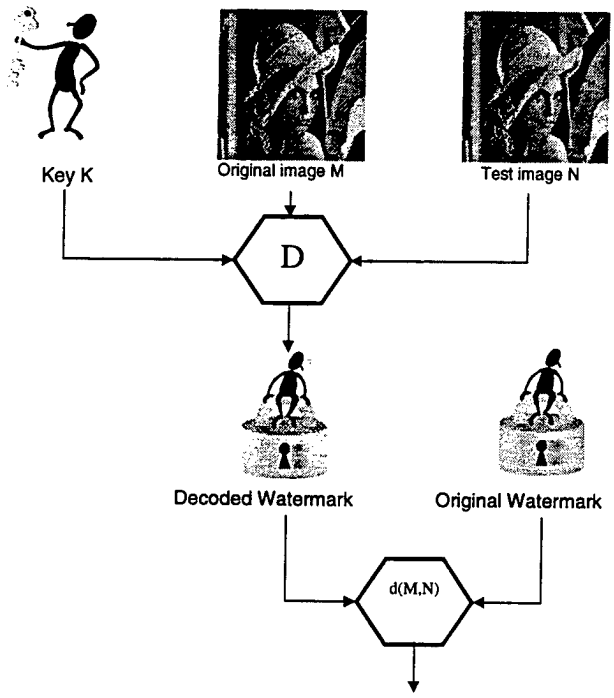


Figure 2. Generalized watermark detection process.

To verify whether an image N is a watermarked version of M, or in other words, to detect the presence of the watermark $V(M)$ in N (**Error! Reference source not found.**), we must check that

$$N - V(M) = M.$$

However, one must take into consideration that it is possible that N may be a modified version of $W(M)$, i.e., $N = F(W(M))$ where F: $\boldsymbol{I}\rightarrow\boldsymbol{I}$ is a certain unknown image processing operation. It is therefore necessary that the detection function d allows partial detection, by taking values close to 1 when N is a partially distorted version of $W(M)$. Normally, we conclude that the watermark $V(M)$ is detected if $d(M,N)$ is greater than a pre-determined threshold.

We will say that the watermarking mapping W is robust with respect to the image processing operation F for the detection function d if the sequence

$$d(M, F^j(W(M))) \qquad j = 0, 1, 2, \ldots$$

decreases slowly from 1 to 0 for every $M \in \mathbf{I}$.

Wavelet transforms

There in an increasing interest in using the wavelet transform to embed watermarks in digital images. One reason for this is that wavelet transformations are computationally cheaper than the commonly-used discrete cosine transform (DCT). The computational complexity of the DCT is $O(n\log n)$ while the discrete wavelet transform is $O(n)$. Another reason for the increase of interest is that the wavelet transform offers a higher degree of security than the DCT. The knowledge that the DCT is used in a watermarking technique gives an attacker an important clue where to mount the attack. However, in the case of wavelet-based watermarking, the wavelet family, the particular member of the chosen wavelet family, and the level of decomposition using the chosen wavelet basis can all be part of a secret key. This key constitutes an additional barrier for an attacker.

Wavelets are a useful tool for representation, approximation and reconstruction of arbitrary functions and in particular for processing data sets like digital images. There exists an extensive literature on the subject and all the wavelet concepts discussed below can be found for example in [D], [M], [STN] and [VK].

A wavelet is a function $\Psi(x)$ in the space $L^2(R)$ of square integrable functions such that the family of functions

$$\Psi_{j,k}(x) = 2^{\frac{j}{2}}\Psi(2^j x - k)$$

is an orthonormal basis in $L^2(R)$, where $j,k \in Z$.

If

$$f(x) = \sum_{j=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty} a_{j,k}\Psi_{j,k}(x)$$

we will call the set of coefficients $\{a_{j,k}\}_{j,k\in Z}$, the discrete wavelet transform (DWT) of the function f. Large values of the index j correspond to high frequency components of f and allow us to represent the details of f in the expansion; the small values of j correspond to low frequency components which constitute a coarse approximation of f.

What distinguishes the wavelet bases among the infinity of possible bases is that:

- wavelets have hierarchical structure (ordered by k in space and by j in frequency/resolution) and form
- a "multiresolution analysis" of increasing spaces span $\{\Psi_{j,k}\}_{k\in Z}$
- wavelets are local in space and in frequency

The multiresolution structure provides a fast wavelet transform whose computation requires only $O(n)$ operations for discrete signals of size n. Wavelets are generated from a single function by scaling and translation and both these operations are directly and efficiently computable.

The space-frequency localization of wavelets allows us to represent a function with a small number of nonzero coefficients in the wavelet expansion which decrease with increasing magnitude of j and k. This is due to the fact that most data, including images, has a certain correlation structure both in space (i.e., nearby pixels are highly correlated while those far apart are uncorrelated) and in frequency (i.e., the spectrum of the image signal is localized in some specific band).

Wavelets are also the building blocks for virtually all common function spaces used in applied mathematics. More precisely, they form an unconditional basis for such function spaces. The spatial localization of wavelets allows one to identify the portion of the image that contributes large wavelet coefficients. The frequency localization of wavelets allows one to identify intensity oscillations in the parts of an image that have sharp variations (e.g., around edges). In contrast, the Fourier transform, which is the standard tool in image processing, is not local both in space and in frequency. A local change in the image affects all the transform coefficients and the coefficients do not (directly) reveal the location of the change.

To any one-dimensional wavelet basis a product (tensor) two-dimensional wavelet basis can be associated by means of taking tensor products of one-dimensional wavelet bases. We will use such bases in analyzing images which are discrete functions of two variables.

Wavelet-based watermarking methods

One of the principal techniques of embedding watermarks in digital images is the transform method. Instead of working directly with the image a certain transform of the image is computed. Watermarking is performed on the

computed transform coefficients and then the inverse transform is applied to the watermarked coefficients to obtain the watermarked image. The first implementations of such wavelet-based methods were given in [KH1, 2].

The general scheme of the wavelet transform watermarking method is given in Figure 3. Given an image M, let $A = A(M,l)$ denote the matrix of the l-th level wavelet approximation coefficients of the image. A specific wavelet-based watermarking method takes A and, using a particular transformation T, generates $T(A)$ which is a watermarked matrix of wavelet coefficients. From $T(A)$, the watermarked image $W(M)$ is constructed. The following two sections describe in detail two distinct watermarking mappings T: the first one constitutes the wavelet-SVD watermarking method the second the wavelet-random additive watermarking method.

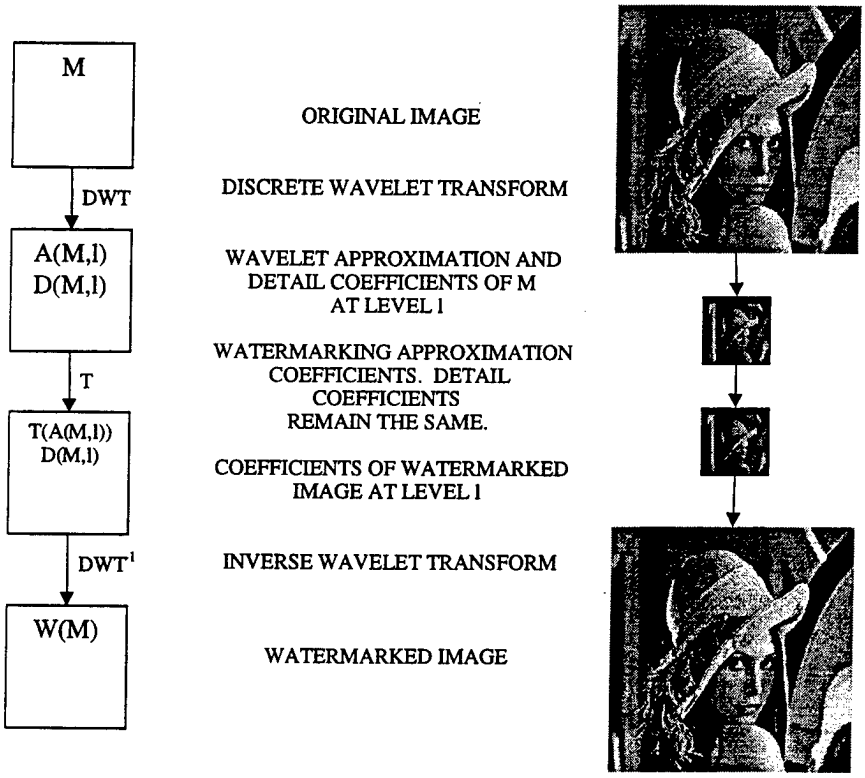


Figure 3. General wavelet-based watermarking method.

Before one can implement the above scheme, one must (1) select a suitable wavelet transform and (2) design the coefficient watermarking transform T.

The task of selecting an optimal wavelet transform for the purpose of watermarking requires a thorough understanding of the transform's analytical properties and of specific characteristics of the class of images under consideration. For any specific task in image processing involving wavelet transformation, it is necessary to consider the following wavelet properties in the wavelet selection: (1) smoothness, (2) order, (3) localization, (4) independence, and (5) symmetry.

Smoothness refers simply to the differentiability of the wavelet function Ψ. A smooth wavelet represents better smooth functions. Increasing the smoothness of the wavelet results in increasing the support of the wavelet and increases the computational complexity of the wavelet transform.

A wavelet is of order m if $\int_{-\infty}^{+\infty} x^k \Psi(x)dx = 0$ for $k = 0,1, ..., m$. A wavelet with high order "suppresses" the polynomial part of the signal f which means that the low frequency wavelet coefficients of f will be almost zero at parts of the domain where f is smooth. High order allows efficient wavelet compression.

Localization is the ability of the wavelet to reveal the discontinuities of f. Also, a wavelet with shorter support will not fuse nearby discontinuities. In other words, the shorter the support, the more localized the wavelet is.

Independence refers to a classification of wavelet bases in the Hilbert space with respect to the cost of reconstructing an element f in the Hilbert space from its Fourier coefficients in the basis. Riesz bases, frames, biorthogonal bases and orthogonal bases are the most often considered types of bases. The cost of reconstruction is the smallest for orthogonal bases.

Finally, symmetric wavelet transforms produce less perceptible approximation errors at the edges of the images. However, increasing the symmetry of the wavelet conflicts with the orthogonality of the wavelet basis.

Once one has selected a suitable wavelet transform, one must design the coefficient watermarking transform T. We propose two watermarking transforms: (1) the wavelet-SVD watermarking method and (2) the wavelet-random additive method.

Wavelet-SVD watermarks

To define wavelet-SVD coefficient watermarking transform T, let $A = A(M,l)$ be the $n \times n$ matrix of approximation coefficients at level l for an image M. Consider the singular value decomposition (SVD) of A which is the following factorization of A:

$$A = \mathrm{U}\,\Sigma\,\mathrm{V}^T$$

where

$$U = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \qquad V = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix}$$

U and V are orthogonal matrices such that $U^T U = I$ and $V^T V = I$.

Let

$$\overline{U} = \begin{pmatrix} \overline{u}_1 \\ \vdots \\ \overline{u}_n \end{pmatrix} \qquad \overline{V} = \begin{pmatrix} \overline{v}_1 \\ \vdots \\ \overline{v}_n \end{pmatrix}$$

be two randomly-generated (possibly key dependent) orthogonal matrices and

$$\overline{\Sigma} = \sigma \begin{pmatrix} \overline{\sigma}_1 & & \\ & \ddots & \\ & & \overline{\sigma}_n \end{pmatrix}$$

be a randomly-generated (also possibly key dependent) diagonal matrix. By replacing last d rows in the matrix U (and respectively in V) by the corresponding rows in the randomly generated matrix $\overline{U}$ (and respectively $\overline{V}$), we define matrices

$$\tilde{U} = \begin{pmatrix} u_1 \\ \vdots \\ u_{n-d} \\ \overline{u}_{n-d+1} \\ \vdots \\ \overline{u}_n \end{pmatrix} \qquad \tilde{V} = \begin{pmatrix} v_1 \\ \vdots \\ v_{n-d} \\ \overline{v}_{n-d+1} \\ \vdots \\ \overline{v}_n \end{pmatrix}$$

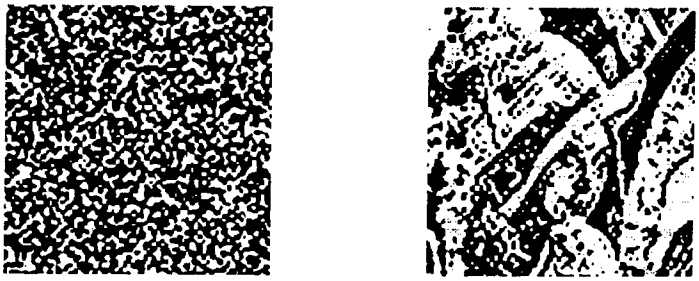
Finally, we define the SVD coefficient watermarking transformation T as

$$T(A) = A+V(A)$$

where

$$V(A) = \tilde{U}\ \tilde{\Sigma}\ \tilde{V}^{\,T}$$

The magnitude of the watermarking modification of the wavelet coefficients of the image is measured by the norm $\|V(A)\|$ and is hence controlled by the size of the scaling parameter σ. The randomness of the coefficient watermark $V(A)=T(A)-A$ is controlled by the parameter d/n. If $d = 0$, the watermark is strongly image dependent; if $d = n$, it is random. Figure 4a gives an example of a highly random wavelet-SVD watermark. Highly random wavelet-SVD watermarks are not highly correlated between each other. Figure 4b illustrates a more image-dependent wavelet-SVD watermark. These watermarks have a higher correlation between each other due to their similarities. Even if d is small, most of the coefficients in A are modified in the course of computing $T(A)$ and consequently the obtained coefficient watermark is dense.



(a) σ= 0.1, d/n = 0.99 (b) σ= 0.3, d/n = 0.40

Figure 4. Wavelet-SVD watermarks W(M)-M.

Figure 5b is an example of an watermarked image using the wavelet-SVD method. The parameters used in watermarking the image are $\sigma = 0.1$, $d/n = 0.99$. Figure 5a is the original image.

(a) Original Image (b) Watermarked Image using Wavelet-SVD

Figure 5. Original and watermarked image.

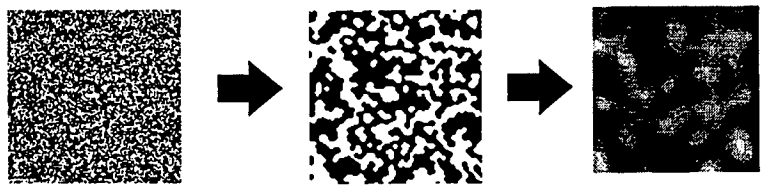
Wavelet-random additive modulation watermark

To define wavelet-random additive coefficient watermarking transform T, we first define a random (possibly key-dependent) matrix $V(A)$ of small norm to be directly added to the wavelet approximation coefficient matrix $A = A(M,I)$. With this, we define the coefficient watermarking transformation as

$$T(A) = A + V(A).$$

The random modulation matrix $V(A)$ should be "strongly" random so that the resulting coefficient watermarks obtained for different seed values are uncorrelated. In constructing $V(A)$, we use the algorithm proposed by J. Fridrich in [F]. The algorithm consists of the following steps:

- Generate a random matrix with entries in the interval $[A_{min}, A_{max}]$ where A_{min}, A_{max} are respectively the smallest and the largest values of the coefficients in A (Figure 6a).
- Use a cellular automaton with a n appropriate voting rule and stopping criterion to coalesce the random pattern in the entries of the matrix generated in the previous step (Figure 6b)
- Apply a smoothing convolution to the matrix generated in the previous step to decrease the size variations of the matrix entries (Figure 6c).
- Subtract the mean of the entries and scale the resulting matrix to obtain an image watermark with entries in the range from –8 to 8.



(a) random matrix (b) coalesced pattern (c) smoothed pattern

Figure 6. Random modulation matrix V(A).

Watermark detection and robustness of wavelet-based watermarks

Any watermarking method includes two basic parts: applying the watermarking mapping and detecting the watermark in the tested image by using specific necessary information about the original image. The previous

section discussed how one can construct the watermarking mapping T. To detect a watermark, let M denote the original image and N denote the image being tested for the presence of the watermark $W(M)$. For any $n \times n$ matrix B, we define $n \times n$ matrix $\hat{B}$ = DCT (B) as a two-dimensional discrete cosine transform of B. As earlier $A = A(M,l)$ denotes the matrix of the l-th level wavelet approximation coefficients. In our wavelet-based watermarking schemes, both watermark embedding and watermark detection depend respectively on the modifications of $A(M)$ and comparisons of $A(M)$ and $A(N)$. For this purpose, we denote

$$V(A(M)) = T(A(M))-A(M)$$

In our watermark detection and robustness testing we evaluated the following detection functions:

$$d_T(M,N)=\frac{\left|\langle V(A(M)),A(N-M)\rangle\right|}{\|V(A(M)\|\|A(N-M)\|} \quad \hat{d}_T(M,N)=\frac{\left|\langle \hat{V}(A(M)),\hat{A}(N-M)\rangle\right|}{\|\hat{V}(A(M)\|\|\hat{A}(N-M)\|}$$

$$d(M,N)=\frac{\left|\langle W(M)-M,N-M\rangle\right|}{\|W(M)-M\|\|N-M\|} \quad \hat{d}(M,N)=\frac{\left|\langle \hat{W}(M)-\hat{M},\hat{N}-\hat{M}\rangle\right|}{\|\hat{W}(M)-\hat{M}\|\|\hat{N}-\hat{M}\|}$$

where $\langle A,B\rangle=\sum_{i,j=1}^{n} a_{ij}b_{ij}$, $\|A\|=\langle A,A\rangle^{1/2}$.

The detection function d_T measures the absolute value of the cosine of the angle between vectors $V(A(M))$ = T(A(M)) – A(M) and $A(N\text{-}M)$ = A(N)-A(M) (Figure 7). In statistical terms, d_T measures the absolute value of the linear correlation between the wavelet approximation coefficients of the watermarked image and the wavelet approximation coefficients of the image under test. Similar geometrical and statistical descriptions can be given to the function d and to its counterparts $\hat{d}_T$ and $\hat{d}$. The DCT-based detection functions are more costly to evaluate, but in most tests they detected the presence of the watermark better than d_T and d.

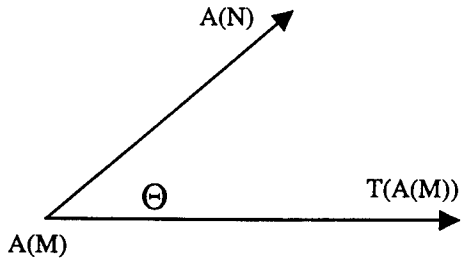


Figure 7. Detection function $d_T(M,N) = |\cos\Theta|$

In our numerical experiments we used a 256x256, 8-bit gray-scale image of Lenna (Figure 8). The basic wavelet used in all watermarking schemes was the orthogonal, compactly supported Daubechies wavelet, referred to as 'db6' in Matlab™, [MMOP]. We selected the wavelet in accordance with general wavelet selection criteria we quoted in the section entitled "Wavelet-based watermarking methods." We tested both watermarking methods at decomposition levels 1 through 5. The strength of the wavelet-random additive image watermark varied in the range from –8 to 8 gray levels. The wavelet-SVD watermark was tested with σ=0.1, d/n=0.99.

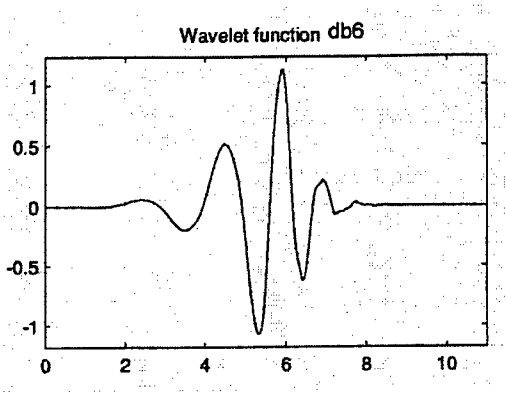


Figure 8. Lenna and wavelet db6.

Based on our experiments, we selected $\hat{d}$ for our watermark detection function. This selection was based on its seemingly better response to certain types of attacks. We are not claiming that this is the optimal detection function for our watermarking schemes. Further research is required to examine other possible detection functions.

Thresholding procedure and detection results

A detection threshold must be set for the detection function before subjecting the watermarking scheme to robustness testing. For a watermarking scheme to be effective, this detection threshold must minimize the number of false detections and non-detections. In order to set the appropriate thresholds for our watermarking methods the original image M was marked with a chosen watermark (watermark seed = 500) and was tested for the presence of 1,000 randomly generated watermarks. Figure 9 shows the results of this experiment for the wavelet-random additive watermarking scheme. Figure Figure 10 shows the results of this experiment for the wavelet-SVD watermarking scheme. In the case of the wavelet-SVD scheme, the chosen watermark was detected with a high detection value (approximately 0.5) while randomly generated watermarks produced smaller values (approximately 0.1) of the detection function $\hat{d}_T$(M, N). To minimize the number of false detections, the threshold must be set above the maximum value of $\hat{d}_T$(M, N) for N ≠ W(M). Experimental results suggest that a threshold of 0.4 should be used for the wavelet-random additive watermark scheme and a threshold of 0.13 should be used for the wavelet-SVD watermark scheme.

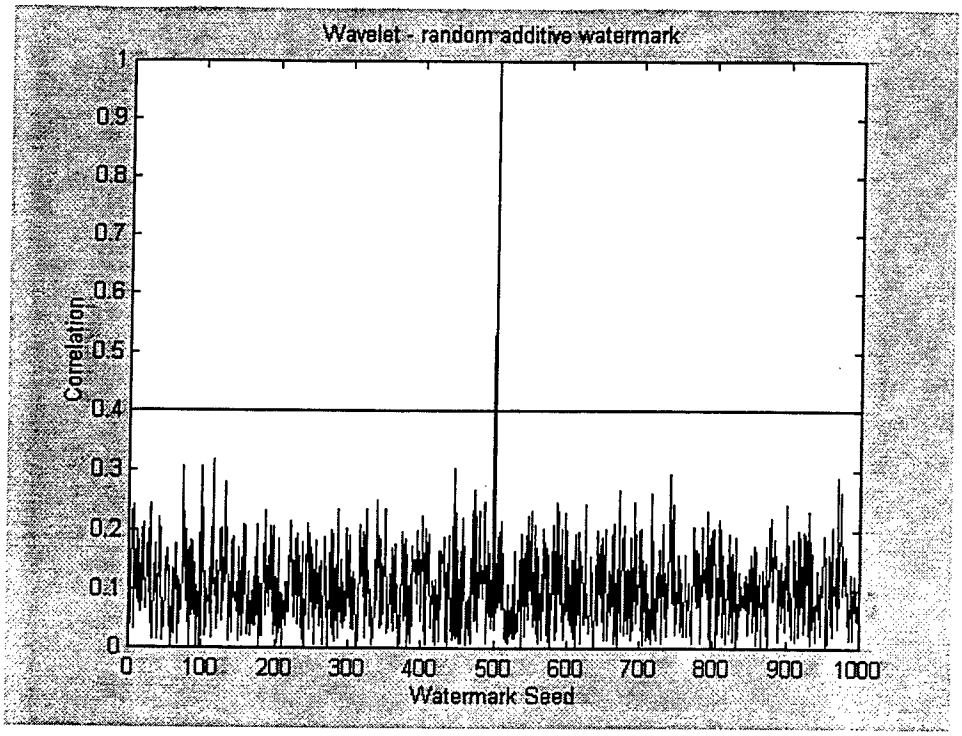


Figure 9. Wavelet – random additive watermark, 'db6' wavelet. Detection of owner's watermark (seed = 500) among 1000 random watermarks and setting the detection threshold $\hat{d}_T$ *=0.4.*

In addition to using the correlation of random watermarks with a chosen watermark to set the threshold, two other experiments were conducted. The first experiment dealt with watermarking the test image at level 4 and measuring its correlation with watermarks at level 1.

Figure 11a shows the results of this experiment. One can see that no $\hat{d}_T$ (M, N) value is greater than our detection threshold. This indicates that the knowledge of the level of wavelet decomposition is important if one wishes to detect the embedded watermark. For security purposes, this knowledge can be incorporated into a key.

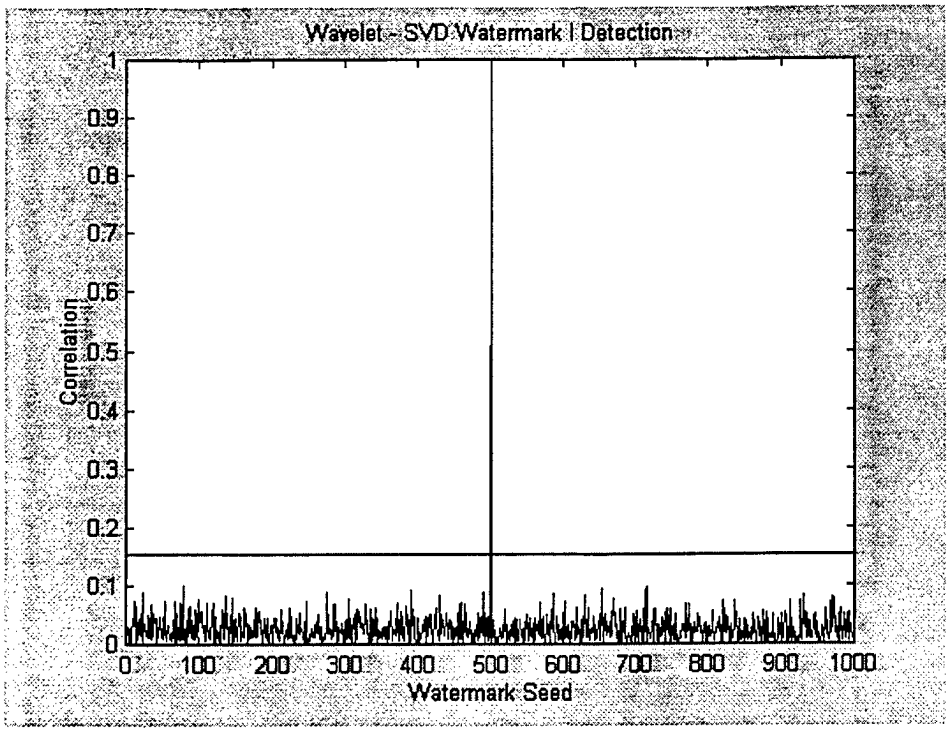
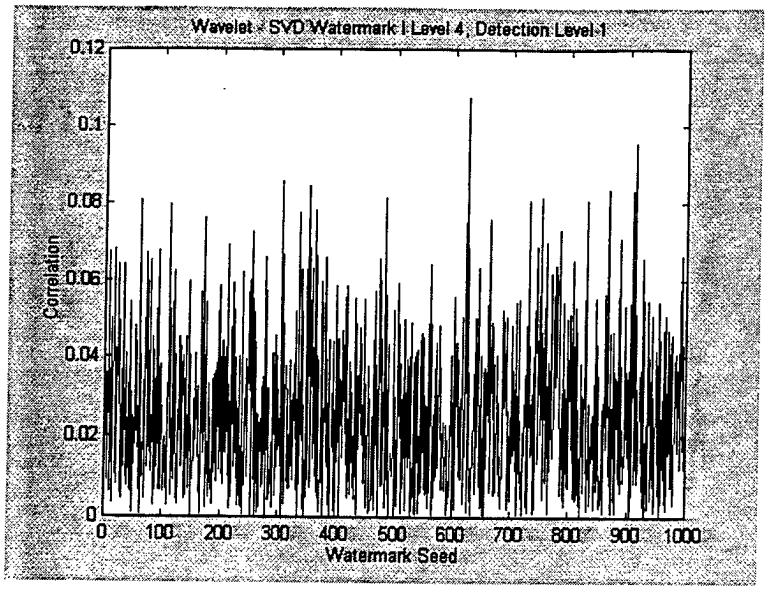


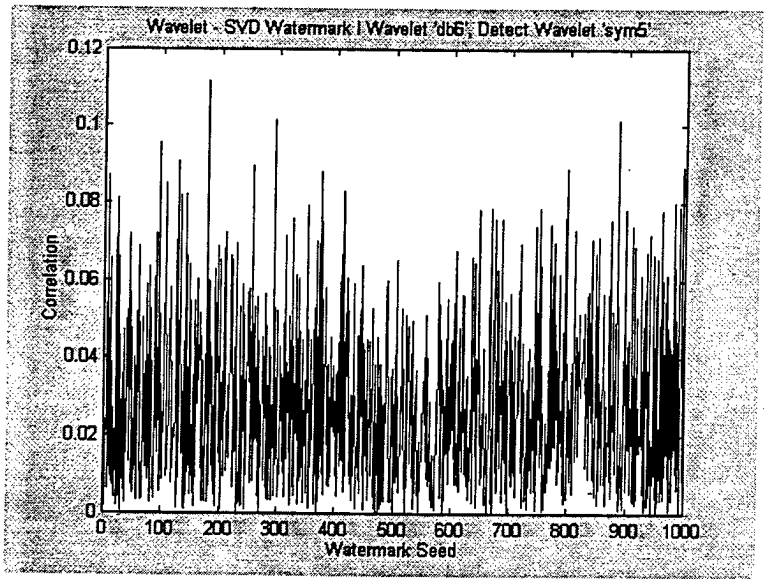
Figure 10. Wavelet – SVD watermark I, 'db6' wavelet. Detection of owner's watermark (seed = 500) among 1000 random watermarks and setting the detection threshold $\hat{d}_T$ *=0.15.*

The second experiment dealt with watermarking the test image using wavelet 'db6' and detecting with wavelet 'sym5'.

Figure 11b shows the results of this experiment. One can see that no $\hat{d}_T$ (M, N) value is greater than our detection threshold. This indicates that knowledge of the specific wavelet basis used in embedding the watermark is important if one wishes to detect the embedded watermark. For security purposes, this knowledge can be incorporated into a key.



(a) Using different levels of wavelet decomposition for watermark and detection (watermark embedded at level 4; detection done at level 1)



(b) Using different wavelets for watermarking and detection (watermark embedded with wavelet 'db6'; detection performed with wavelet 'sym6')

Figure 11. Watermark detection failure using incompatible wavelet transforms for watermarking & detection.

Robustness tests and results

All tests were done using Matlab™ by The MathWorks, Inc. These Matlab™ test scripts were initially implemented by Dr. Jiri Fridrich of SUNY Binghamton and were adapted by John Vergis and 2Lt Arnold Baldoza of the Air Force Research Laboratory.

All of the robustness tests are parameterized: for example, in the blurring test, we vary the number of times the blurring kernel is iterated over the image; for the JPG compression test, the quality factor is varied between zero and

100. At each step, the watermark is detected and the corresponding value of the detection function is noted. If the detection value is above the appropriate threshold, the watermark is said to be persistent at the level of testing. If the detection value is below the appropriate threshold, the watermark is considered to be undetectable.

Error! Reference source not found. The following tables (Table 1 and Table 2) present the results of the robustness tests for the wavelet-random additive watermark and the results of the robustness tests for the wavelet-SVD watermark.

Table 1. Results of robustness tests for the wavelet-random additive watermark.

Test	Detection Function			
	d_T	$\hat{d}_T$	d	$\hat{d}$
JPG Quality (quality units)	10	4	40	4
Blurring (kernel iterations)	0	7	0	7
Median Filtering (block size)	3	11	1	12
Noise Adding (% of uniform noise)	10	30	5	30
Mosaic Filtering (block size)	2	7	1	7
Gamma Correction (gamma exponent interval)	[0.9,1.1]	[0.5,1.1]	[0.9,1.1]	[0.7,2.0]
Pixel Permutation (block size)	2	7	1	7

Table 2. Results of robustness testing for wavelet-SVD at Level 1 with $\sigma = 0.1$ and d/n = 0.99.

Test	Detection Function			
	d_T	$\hat{d}_T$	d	$\hat{d}$
JPG Quality (quality units)	17	15	15	15
Blurring (kernel iterations)	0	0	1	1
Median Filtering (block size)	3	4	3	4
Noise Adding (% of uniform noise)	30	15	15	15
Mosaic Filtering (block size)	2	2	3	3
Gamma Correction (gamma exponent interval)	(0, 1.1]	[0.9, 1.1]	(0, 1.0]	[0.9, 1.3]
Pixel Permutation (block size)	2	1	3	3

Blurring procedure and results

In performing robustness tests with respect to blurring, the blurring kernel of PaintShop Pro 4.1 (Figure 12) was used.

$$1/44 \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 2 & 2 & 8 & 2 & 2 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

Figure 12. Blurring kernel.

Based on the appropriate thresholds, the wavelet-random additive watermark scheme withstands 7 blurring kernel iterations and the Wavelet-SVD watermark scheme I scheme endures 1 blurring operation. Figure 13 shows the watermarked images at these extreme blurring levels.

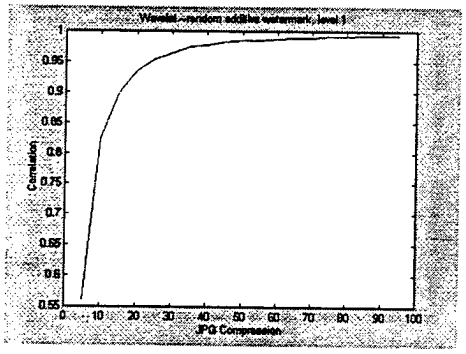
Wavelet-random additive watermark

Wavelet-SVD watermark

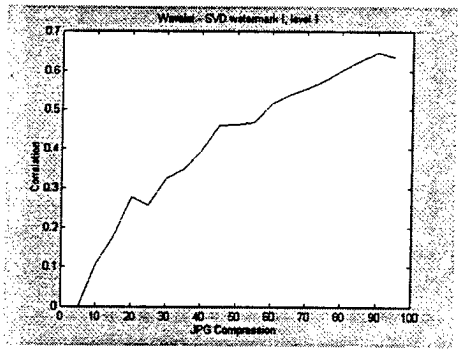
Figure 13. Watermarked images at blurring detection threshold.

JPEG compression procedure and results

The 256x256 image of Lenna was watermarked by both our watermarking methods and the resulting watermarked images were subjected to various levels of JPG compression. Figure 14 shows the robustness performance of each watermarking technique with respect to these compression levels.



(a) Wavelet-random additive watermark



(a) Wavelet-SVD watermark

Figure 14. Robustness performance with respect to JPG compression.

Based on the established thresholds, a watermark embedded by the wavelet-random additive watermark scheme can be detected in the image even after being JPG encoded with a quality factor of 4. A wavelet-SVD watermark persists until a quality factor of 15 is reached. Figure 15 shows the watermarked Lenna images at these quality factors.

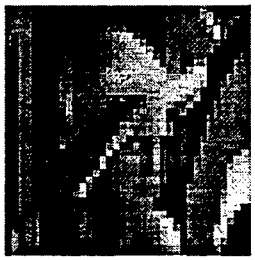
Wavelet-random additive watermark

Wavelet-SVD watermark

Figure 15. Watermarked images at JPG compresion detection threshold.

Mosaic filtering procedure and results

Mosaic filtering divides the image into square subblocks and replaces each block by its average gray level. The wavelet-random additive watermark scheme survives block sizes of 7; the wavelet-SVD watermark persists with block sizes of 3. Figure 16 shows the watermarked Lenna images at these block sizes.



Wavelet-random additive watermark

Wavelet-SVD watermark

Figure 16. Watermarked images at mosaic filtering detection threshold.

Gamma correction procedure and results

Gamma correction raises every entry of the image to power gamma, [TS]. The wavelet-random additive watermark scheme can be detected when gamma is in the interval [0.7, 2.0]; the wavelet-SVD watermark persists in [0.9, 1.3]. Figure 17 shows the performance of these watermarking techniques with respect to gamma correction.

$\gamma = 0.7$

$\gamma = 2.0$

Wavelet-random additive watermark

$\gamma = 0.9$

$\gamma = 1.3$

Wavelet-SVD Watermark

Figure 17. Watermarked images at gamma correction detection thresholds.

Pixel permutation procedure and results

Pixel permutation divides the image into square subblocks and permutes the pixels in each of these subblocks. The wavelet-random additive watermark scheme withstands permutations of block sizes of 7; the wavelet-SVD watermark persists with block sizes of 3. Figure 18 shows the watermarked Lenna images at these block sizes.

Wavelet-random additive watermark

Wavelet-SVD watermark

Figure 18. Watermarked images at pixel permutation detection thresholds.

Median filtering procedure and results

Median filtering divides the image into square subblocks and replaces each pixel by the median value of its subblock. In this test, one assumes that the image is padded with zeros outside its boundaries. The wavelet-random additive watermark scheme survives block sizes of 12; the wavelet-SVD watermark persists with block sizes of 4. Figure 19 shows the watermarked Lenna images at these block sizes.

Wavelet-random additive watermark

Wavelet-SVD watermark

Figure 19. Watermarked images at median filtering detection thresholds.

Noise adding procedure and results

The noise adding test adds Gaussian white noise of mean 0 and with standard deviation of amplitude/256, amplitude is varied from 5 to 95 at increments of 10. The wavelet-random additive watermark scheme survives noise adding until an amplitude of 30; the wavelet-SVD watermark persists until noise amplitudes of 15. Figure 20 shows the watermarked Lenna images at these noise levels.

Wavelet-random additive watermark

Wavelet-SVD watermark

Figure 20. Watermarked images at noise adding detection thresholds.

Analysis of robustness tests for wavelet-random additive and wavelet-SVD watermarks

Comparing our watermarking methods is a delicate and difficult problem. This is a task difficult to formulate and rigorously resolve. Watermarking algorithms can be compared according to some basic criteria. Examples of these criteria are

- numerical cost;
- information about the original image used to generate the watermark;
- strength of the watermarked measured in a matrix norm or in terms of a number of embedded bits;
- density measured in terms of the number of nonzero entries of the watermark;
- watermark perceptibility;
- watermark robustness to specific image processing operations (filters).

Without being able to rigorously assert that the algorithms under considerations are "equal" with respect to a specific criterion it is meaningless to compare them with respect to any other criteria.

In the case of our two wavelet-based methods we can assert that the information $A(M,l)$ about the original image used by both methods is the same. Both methods generate imperceptible dense watermarks and perform well at levels $l = 1$ and 2. Wavelet-SVD watermark continues to perform well on the coarse levels up to $l = 5$ (Figure 22) whereas the detection of the random additive watermark falters [illegible]. Wavelet-SVD watermark offers more precise control on the strength of the generated watermark. It also allows one to control the degree of randomness of the watermark from almost completely random (wavelet-SVD watermark with $\sigma = 0.1$, $d/n = 0.99$) to strongly image dependent watermark (wavelet-SVD watermark with $\sigma = 0.1$, $d/n = 0.4$). However, when $d/n = 0.4$, the threshold to minimize false detection is high, causing the watermark to be less robust to image processing operations.

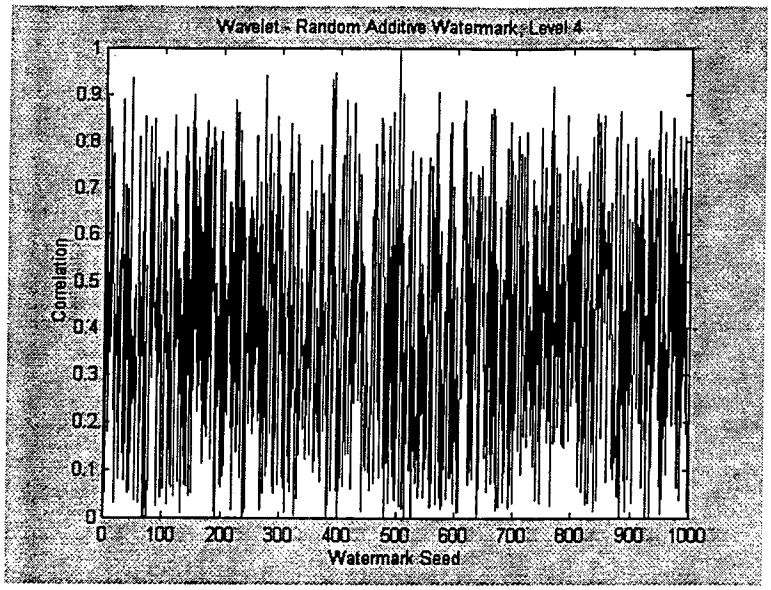


Figure 21. Wavelet – random additive watermark, wavelet 'db6' level 4 – Correlation with random watermarks.

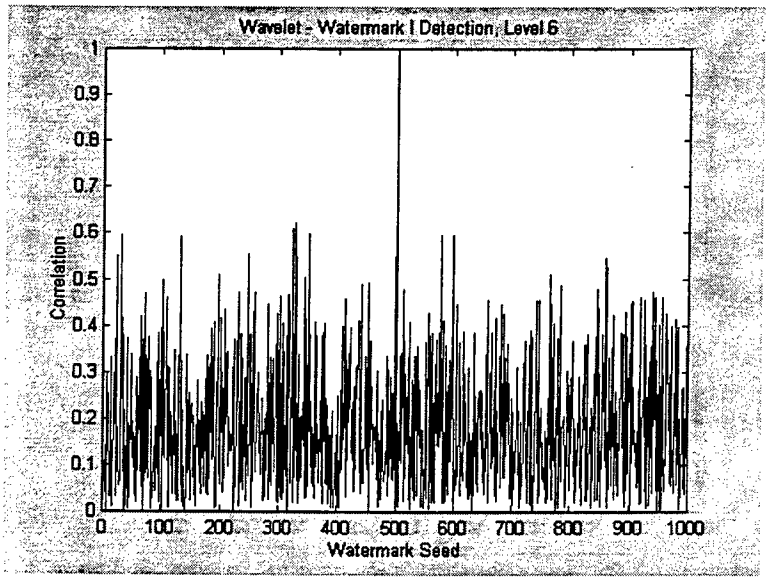


Figure 22. Wavelet – SVD watermark I, 'db6' wavelet level 6 – Correlation with random watermarks.

Wavelet-SVD watermarking algorithm worked faster requiring on average 108 cyc/byte (or 3.5 sec on Gateway 2000 G6-200 with Pentium Pro processor) to embed and respectively 271 cyc/byte (8.7 sec) to detect. Wavelet-random additive watermarking embedded in 125 cyc/byte (4.1 sec) and detected in 288 cyc/byte (9.4 sec).

Conclusions

There are two main reasons for using the wavelet-based watermarking methods. The first reason is its computational cost of $O(n)$ operations as oposed to the discrete Fourier transform methods, which requires $O(n \log$

A Model To Analyze Sensor Data For Detection of Multi-Source Attacks

Brajendra Panda
Assistant Professor
Department of Computer Science

University of North Dakota
P.O. Box 9015
Grand Forks, ND 58202-9015

Final Report
for
Summer Faculty Research Program
Rome Research Site

Sponsored by:

Air Force Office of Scientific Research
Bolling Air Force Base
Washington, D.C.
and
Rome Research Site
Rome, NY 13440

August 1998

A Model To Analyze Sensor Data For Detection of Multi-Source Attacks

Brajendra Panda
Assistant Professor
Department of Computer Science
University of North Dakota

Abstract

Sensors at different Air Force operation sites collect information on various network and system parameters and store them in local databases. Data obtained from various are then analyzed to detect any occurrence of attacks. If any indication of an attack is found, necessary course-of-action is taken to recover the system(s) affected by the attack. While copycat types of attacks can be easily detected, more prudent coordinated attacks originating from multiple sites by a group of sophisticated attackers will go unnoticed by sensors. This research presents a model to detect such attacks. It uses Petri Nets to represent the model. We claim that this model will significantly reduce number of false alarms raised by the intrusion detection sensors.

A Model To Analyze Sensor Data For Detection of Multi-Source Attacks

Brajendra Panda

1. Introduction

With the availability of cheap but powerful computers and easy access to Internet, collection and dissemination of information through electronic media has become a common practice. However, this practice also involves a considerable amount of risk – disclosure or even loss of sensitive information through unauthorized access to one's computer system by malicious intruders. A recent study conducted by the President's Commission on Critical Infrastructure Protection [PCCIP97] found that today's computer infrastructure is vulnerable to attacks. In this report, Mr. Robert Marsh, the chairman of the committee, states:

> *"The capability to do harm – particularly through information networks – is real; it is growing at an alarming rate; and we have little defense against it."*

Attack on an organization's information resources through electronic media is termed as information warfare. The objective of an attacker could range from causing momentary disorder in the organizational activities to complete destruction of the opponent's information system. Defensive Information Warfare (DIW) prepares the system to withstand such attacks while providing system integrity at all time. DIW plays a vital role in the protection of command, control, communications, and computer (C^4) systems of US Air Force (USAF) [Metc97]. Three major aspects of DIW are: protection of the system from outsiders, detection of an intrusion, and recovery from any damage caused by attackers.

Significant amount of research, including strict access control mechanisms, has been performed to protect systems from unauthorized users. Some of the related research done by the investigator can be found in [Perr93], [Pand94], [Pan95a], and [Pan95b]. Ammann et al. [Amm97] and Graubart et al [Grau96] have discussed some of these issues with regard to information warfare environment. There exist various intrusion detection protocols, which watch suspicious activities of users - both insiders and outsiders - and detect any abuse. McDermott and Goldschalg have developed a method called storage jamming, [Mcd96a] and [Mcd96b] that uses fake data to attract attackers to update the values. Regular users do not use these data, and values of such data are predetermined. Any change to these values

confirms the activity of an intruder. Likewise, audit trails can be established to hold authorized users accountable for their activities and identify any misuse of privileges. A statistical approach to intrusion detection has been discussed in [Lunt90]. In [Lunt93] a survey of various intrusion detection techniques has been provided.

Before discussing the objective of this research, we need to consider the status of current information protection capabilities employed by the USAF. Different Air Force operation sites have sensors to monitor network accesses – both outgoing and incoming traffic. These sensors continually collect data on various system and network parameters and then store the collected data to be analyzed by human experts. This process is highly time intensive and therefore inefficient. As suggested by the [Metc97], more automated processing is needed for analyzing and suggesting a course-of-action when an intrusion is detected. It is desirable to have a decision support system in place to perform the task of human experts. A suggested DIW architecture is depicted in Figure 1.

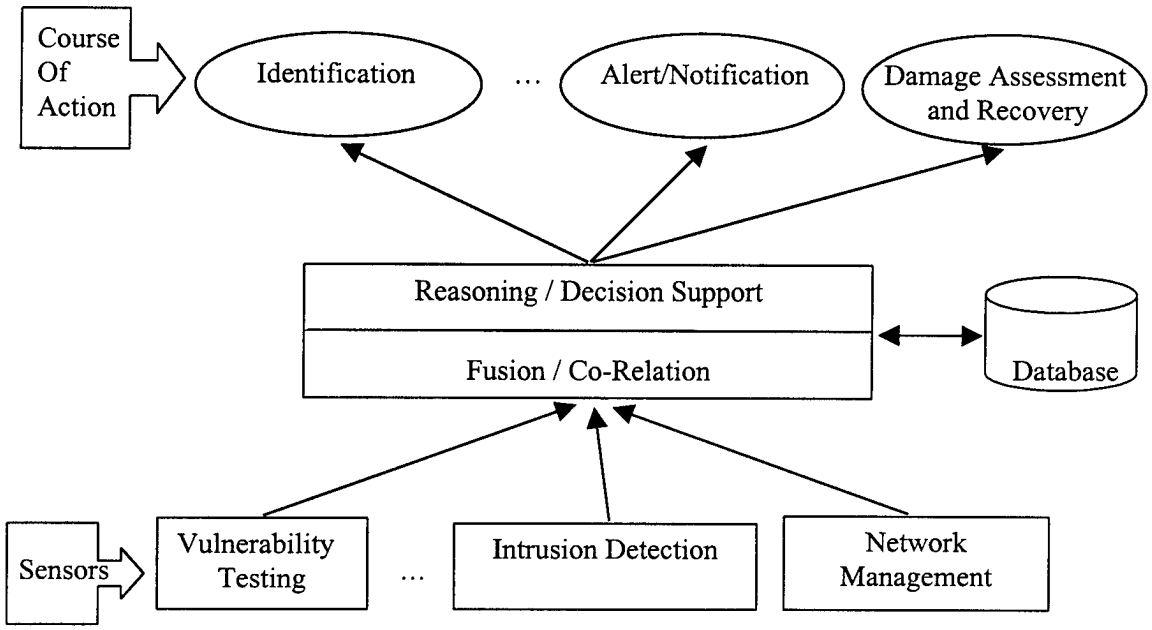


Figure 1: Defensive Information Warfare Architecture

A system called Extensible Prototype for Information Command and control (EPIC) is currently under development at Rome Research Site for providing defensive information warfare decision support.

An architectural view of EPIC is presented in figure 2. Data obtained from various sensors at this site are passed through an expert system called G2, which then stores essential data in a database. Examples of information collected and stored in the database include user ids and passwords, login time, logout time, remote system id (from where access was made), id of the system that was accessed, commands executed, etc. If any suspicious activity by a user is found, the expert system will alert the security administrators of the event. Such activities are usually determined by examining various commands executed by users. Using the current approach, it would be easy to detect copycat types of attacks launched by a single attacker. However, any coordinated attacks organized by a group of sophisticated attackers will be impossible to detect without an efficient, optimized, and refined process that requires automated correlation of data obtained from various sensors.

In this research, we have developed a method for analysis of sensor data to determine attacks that are launched from several distributed sites – may be organized by a group of attackers or by a single attacker who uses his access to systems at several other sites to launch the attack. In section 2, we have presented the research model. We have used Petri Nets in representing this prototype. A brief overview of Petri Nets is discussed in section 3. In section 4, we have described how Petri Nets can be used as a tool to portray our model. Section 5 offers the conclusion of this research.

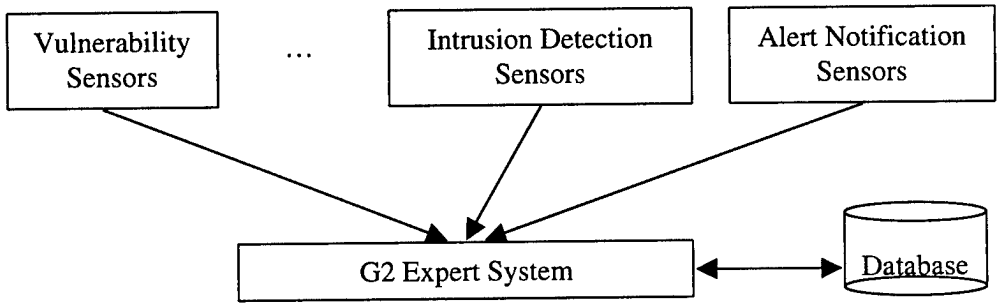


Figure 2: The EPIC Architecture

2. The Model

This research focuses on the commands typed by users to check for any inconsistencies and/or suspicious activities. For this purpose, we analyze commands typed by users for attack patterns. For this purpose, several known attack patterns are collected and stored in the database. As new patterns unfold they can be added to the set of existing patterns. These patterns consist of a series of commands and can

be represented by a graph. A node in the graph signifies a command. An order between two commands is denoted by an edge, which points to the command that must follow the other command where the edge begins. If two commands can be executed independently, i.e., without any particular order, then there would be no edge connecting the corresponding nodes. We explain several of these possibilities using the following figures.

Figure 3 (a) depicts an attack launched by executing two commands, C_1 and C_2, and there is no ordering between C_1 and C_2. Therefore, the attacker may execute them in the order C_1C_2, or C_2C_1. However, both commands must be executed to launch the attack.

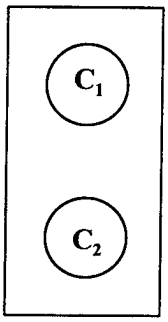


Figure 3 (a): An Attack Pattern with Two Independent Commands

An attack launched by executing the same two commands, C_1 and C_2, where C_1 must be executed prior to C_2 is shown in Figure 3 (b). If the database reveals that both C_1 and C_2 have been executed but the order between them is C_2C_1, we must ignore them since the sequence of operations is not considered as an attack.

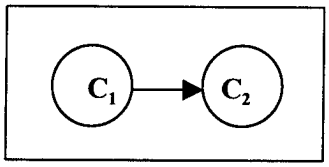


Figure 3 (b): An Attack Pattern with a Required Order Between Two Commands

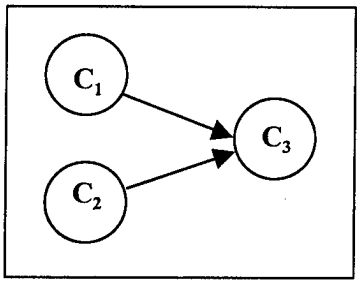


Figure 3 (c): An Attack Pattern Constituting a Partial Order Among Three Commands

Figure 3 (c) displays a case when there are three commands C_1, C_2, and C_3, and the requirement for execution is that C_3 must be executed only after both C_1 and C_2 are executed. However, there is no required order between C_1 and C_2. Therefore, the orders $C_1C_2C_3$, and $C_2C_1C_3$ raise alarm while $C_1C_3C_2$, $C_2C_3C_1$, $C_3C_1C_2$, and $C_3C_2C_1$ are ignored.

Another pattern using a different partial order among three commands, C_1, C_2, and C_3, are shown in Figure 3 (d). In this setting, C_1 is to be executed before the other two. Sequences to be searched are $C_1C_2C_3$ and $C_1C_3C_2$. Other sequences do not constitute the attack.

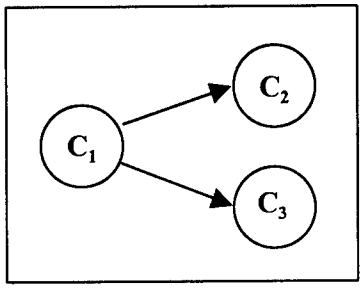


Figure 3 (d): A Different Attack Pattern Constituting a Partial Order Among Three Commands

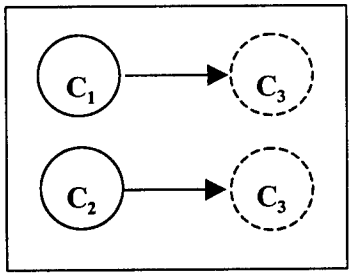


Figure 3 (e): An Attack Pattern Showing a Disjunction Among Three Commands

There may be a case when a command must follow one of the two other commands. This case is displayed in Figure 3 (e). Command C_3 must be executed after execution of C_1 or C_2, and there is no relationship between C_1 and C_2. A sequence of C_1C_3 or C_2C_3 must be searched in this case to detect the attack. However, since either C_1C_3 or C_2C_3 can cause the damage, for simplicity we represent the pattern as two graphs as shown in Figure 3 (f). Similarly, the case when a command must precede either of two other commands can be resolved, and we do not exhibit the figures here.

Using these notations, we can take any attack pattern and use the sequence of commands to build a graph for the attack. Let us consider an attack where command C_3 is executed after execution of C_1

and C_2 with no ordering bewteen C_1 and C_2. Then commands C_5, C_6 and C_7 must be executed in any order, however, C_7 requires that it be executed only after C_4 is executed. Finally, execution of command C_8 completes the attack. By using these sequences we can then draw the graph as shown in Figure 4.

Using these notations, we can take any attack pattern and use the sequence of commands to build a graph for the attack. Let us consider an attack where command C_3 is executed after execution of C_1 and C_2 with no ordering bewteen C_1 and C_2. Then commands C_5, C_6 and C_7 must be executed in any order, however, C_7 requires that it be executed only after C_4 is executed. Finally, execution of command C_8 completes the attack. By using these sequences we can then draw the graph as shown in Figure 4.

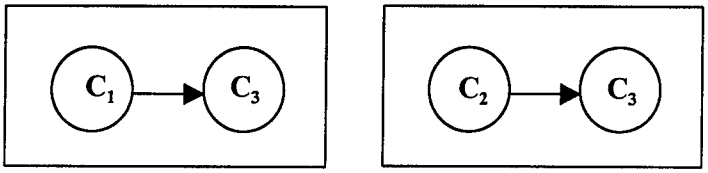


Figure 3 (f): Breaking The Disjunction Type Graph Shown in Figure 3 (e) Into Two Graphs

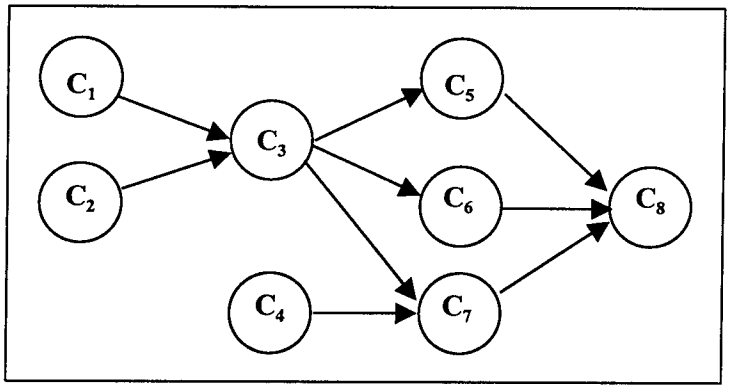


Figure 4: A Graph Representing A Complete Attack Sequence

After building all these attack patterns, our next job is to utilize these graphs to detect an attack. To do this, we scan the database that stores sensor data to find any of these commands. If at least one of the commands is found, we must try to find the existence of the rest of the commands. If few of the earlier commands are not executed, then there is nothing to worry. If the missing commands are the later commands in the sequence and could be executed to complete the attack, then the log must be checked continuously for appearance of these commands. In the mean time, the system must also detect the user/system that is executing the commands and notify security personnel of the event. However, if all commands are found, then we check to see if the partial order among them matches with that of the

graph. If they do match, the occurrence of attack is reported. If they do not match, we declaire that there was no attack. These steps are provided in the algorithm below. The algorithm is run whenever a command belonging to any of the attack graphs is found in sensor data.

Algorithm:

Input: Command C_i, which is a node in each graph in the set of graphs $G = \{G_1, G_2, \ldots, G_n\}$.

1. For each graph $G_j \in G$, search sensor data for existence of other commands in G_j

    1.1 If all commands in G_j are found in sensor data

        Match their execution partial order with that of G_j;

        1.1.1 If the order matches

            Notify of attack G_j;

            Go to Step 1; /* Try next graph in G */

        1.1.2 Else

            Go to Step 1; /* Try next graph in G */

    1.2 Else

        For any two commands C_p and $C_q \in G_j$ such that C_p appears before C_q in G_j

        1.2.1 If C_q is found in sensor data where as C_p is missing /* order mismatch */

            Go to Step 1; /* Try next graph in G */

        1.2.2 Else

            Recheck the log later for missing commands to appear;

            Go to Step 1; /* Try next graph in G */

Advantages Of Our Model:

Our model offers the following advantages.

1) It eliminates false alarms. Existing systems look for the cautionary commands in the log of sensor data, and if any found, they calculate a risk-level associated with each command. If the total risk-level for all the cautionary commands executed by a user exceeds a pre-determined threshold, the system raises the alarm of possible attack, even though the sequence of commands do not match any attack patterns, or there is no co-relation among the commands.

2) This model positively identifies all known attacks. By matching the sequence of commands with the graph, the model guarantees the occurrence of the attack.

3) Most importantly, our model can identify multi-source attacks. Instead of looking for commands executed by individual users, the model looks for attack patterns in the entire database containing sensor data. If any found, the model could establish a relationship among various users/systems that executed the sequence of commands.

4) It offers greater modeling power. As new attack patterns are understood, they can be added to the existing set of graphs. Users can also assign a weight function to each node (or each edge) in the graph to determine the severity of execution of a particular command (or executing two commands in a sequence). Using this process a threshold can be determined for incomplete attacks from which the affected system needs to be recovered.

3. An Overview of Petri Nets

We use Petri Nets to represent our model. In this section, we present a brief overview of Petri Nets. Interested readers are referred to [Mura89].

A Petri Net (PN) is a modeling tool used to study information processing systems that are concurrent, distributed, parallel, non-deterministic, and asynchronous in nature. A PN is represented by a directed, weighted, bipartite graph with two types of nodes, called *places* and *transitions*, where edges are either from a place to a transition or from a transition to a place. Places are drawn as circles and transitions are drawn as bars or boxes. Places can have *markings* (also known as *tokens*), and edges can have *weights* assigned to them. A formal definition of a PN is presented below.

Definition [Mura89]: A Petri Net is a 5-tuple, PN = (P, T, F, W, M_0) where

$P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places,

$T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (edges),

$W: F \rightarrow \{1, 2, 3, \ldots, \}$ is a weight function,

$M_0: P \rightarrow \{0, 1, 2, \ldots\}$ is the initial marking,

$P \cap T = \varnothing$ and $P \cup T \neq \varnothing$.

A transition takes a PN from one state to another state, thus simulating the dynamic behavior of the system under investigation. A transition t is enabled (*fired*) when each of its input places p has $w(p, t)$ tokens, where $w(p, t)$ is the weight of the edge from p to t. After the firing, from each input place p of t, $w(p, t)$ tokens are removed, and $w(t, p)$ tokens are added to each of t's output place p, where $w(t, p)$ is the weight of the edge from t to p. It must be noted that the weight of an edge when not mentioned in the graph is considered as 1.

Figure 5 (a) depicts an example of a simple Petri Net in which there are five places p_1, p_2, p_3, p_4, and p_5, and two transitions t_1 and t_2. The input places for t_1 is p_1 and p_2. The output place for t_1 is p_3. Similarly, p_3 and p_4 are input places for t_2, and p_5 is the output place for t_2. Since for transition t_1 to fire it requires one token for p_1 and two tokens for p_2, and p_1 and p_2 have the required tokens, the transition t_1 takes place. Then, one token from p_1 and two from p_2 are removed, and a token to p_3 is added. The result is shown in Figure 5 (b). However, transition t_2 can not occur, as p_4 does not have a token.

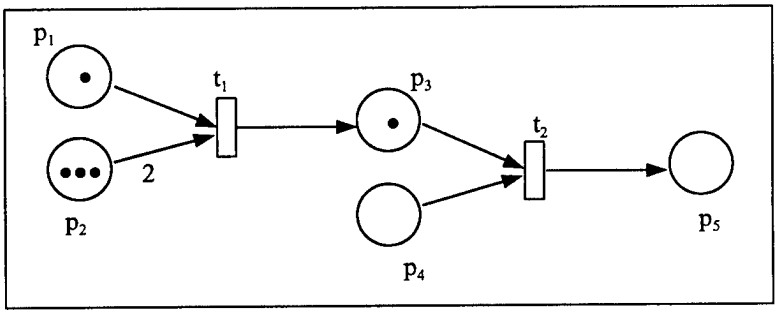


Figure 5 (a): A Simple Petri Net

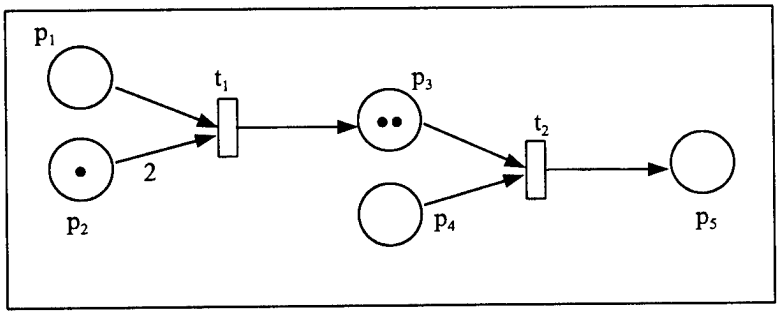


Figure 5 (b): The Petri Net After Transition t_1 Occurs

4. Representing Our Model Using Petri Nets

Since the attack scenario under investigation is concurrent, parallel, distributed, non-deterministic, and asynchronous, it is suitable to use PNs to represent our model. Each node in our attack graph can be denoted as a place in the PN. The partial order in our graph remains the same in the PN. For every edge in our graph, there will be a transition in the PN. Finally, there will be an extra place as the last node in the PN to denote the completion of attack, i.e., a token in this place emphasizes that a required sequence of commands for the attack has been completed. Whenever a command is executed, a token is added to the corresponding place. In order to guarantee that a transition occurs only when the commands are executed in the required sequence, we add the following constraints to the usual transition rule. The tokens appear in two colors, red and blue, for example. When a command is executed a blue token is added to the corresponding place, and when a firing takes place red token(s) are added to the output place(s) of the transition. If a place p_j has a blue token and then a transition t_i occurs which adds a red token to p_j, we call it a *misfire*, and remove the newly added red token from p_j in addition to removing the blue token(s) from the input place(s) of t_i. This deduces that a command (corresponding to the place p_j) was executed prior to execution of command(s) that caused the transition t_i to fire, and, therefore, was out of sequence.

Figure 6 illustrates a PN for the attack sequence shown in Figure 4. In order to explain how a transition would take place, let us consider the following two cases.

Case One: Assume that commands C_1 and C_3 have been executed in the order C_1C_3. By looking at Figure 4, we know that attack indicating sequences are $C_1C_2C_3$, and $C_2C_1C_3$. Using the PN shown in Figure 6, places p_1 and p_3 will have a blue token each. Since p_2 does not have a token, t_1 would not fire. Even if the command C_2 is executed later thus putting a blue token in p_2, this will cause t_1 to misfire. Therefore, p_1 and p_2 will loose their tokens, and p_3 will not get the red token. So, p_3 will never have two tokens required for t_2 to fire (note that the edge from p_3 to t_2 has a weight of 2). Hence, transitions t_4 and t_5 will never fire and p_9 will, therefore, not have a token. This will indicate that the attack was not real.

Case Two: If commands C_1, C_2, and C_3 are executed in the order $C_1C_2C_3$, for example, this will cause p_1, p_2, and p_3 to have a blue token each. Before p_3 gets the blue token, t_1 will fire adding a red token to p_3 and removing the blue tokens from p_1 and p_2. When C_3 is executed, p_3 acquires the blue token, and

this will cause t_2 to fire. If all the required commands are executed in proper sequence, finally, p_9 will have a red token obtained by the firing of t_5. This will indicate the occurrence of the attack.

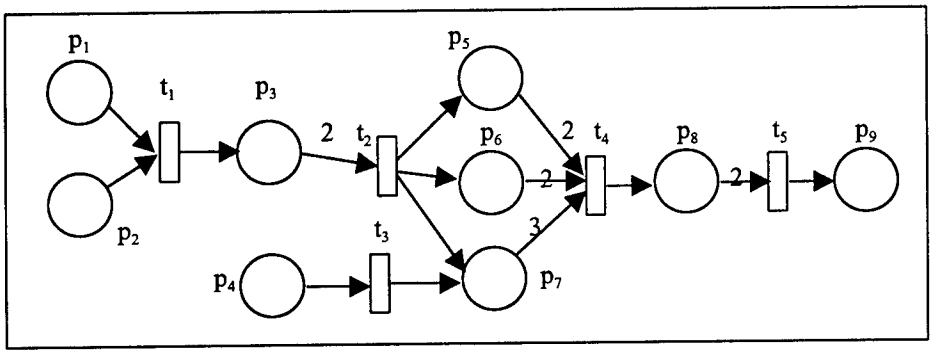


Figure 6: A Petri Net Representing The Attack Graph of Figure 4.

5. Conclusions

In this research, we have developed a model, which uses sensor data to detect attacks that are multi-source in nature. This model is based on a graphical approach. Nodes on the graphs are the commands in the attack sequence, and edges between two nodes represent the execution order between the two corresponding commands that must be observed in order for the attack to take place. By using all the partial orders among these commands, our model determines if the attack ever occurred. This significantly reduces the number of false alarms raised by various intrusion detection systems. We have shown that Petri Nets can be used to represent our model. This offers better modeling capability and also will make the implementation more convenient, if we decide to implement the model.

Acknowledgments

This work was made possible by the summer research program of AFOSR. I sincerely acknowledge, and thankful for, the support of Mr. Joseph Giordano of Rome Research Site during this research. I also thank Mr. Chet Hosmer of WetStone Technologies for his valuable suggestions regarding this work.

References

[Amm97] P. Ammann, S. Jajodia, C. D. McCollum, and B. Blaustein, "Surviving Information Warfare Attacks on Databases", In Proceedings of the 1997 IEEE Symposium on Security and Privacy, p.164-174, Oakland, CA, May 1997.

[Grau96] R. Graubart, L. Schlipper, and C. McCollum, "Defending Database Management Systems Against Information Warfare Attacks", Technical report, The MITRE Corporation, 1996.

[Lunt90] T. F. Lunt, "Using Statistics to Track Intruders", In Proceedings of the Joint Statistical Meetings of the American Statistical Association, August 1990.

[Lunt93] T. F. Lunt, "A Survey of Intrusion Detection Techniques", Computers & Security, Vol. 12, No. 4, p. 405-418, June 1993.

[Mcd96a] J. McDermott and D. Goldschlag, "Storage Jamming", Database Security IX: Status and Prospects, D. Spooner, S. Demurjian, and J. Dobson, editors, p. 365-381, Chapman & Hall, London, 1996.

[Mcd96b] J. McDermott and D. Goldschlag, "Towards a Model of Storage Jamming", In Proceedings of the IEEE Computer Security Foundations Workshop, p. 176-185, Kenmare, Ireland, June 1996.

[Metc97] Therese R. Metcalf, "Computer Security Assistance Program For The Twenty-First Century (CSAP21) Functional Requirements – Draft", Technical Report, The MITRE Corporation, Bedford, MA, December 1997.

[Mura89] Tadao Murata, "Petri Nets: Properties, Analysis and Applications", In Proceedings of the IEEE, Vol. 77, No. 4, p. 541-580, April 1989.

[Pand94] B. Panda, W. Perrizo, and R. Haraty, "Secure Transaction Management and Query Processing in MLS Database Systems", In Proceedings of 1994 ACM Symposium on Applied Computing, pp. 363-368, Phoenix, AZ.

[Pan95a] B. Panda and W. Perrizo, "Query Execution in PRISM and SeaView: A Cost Analysis", In Proceedings of 1995 ACM Symposium on Applied Computing, San Jose, CA.

[Pan95b] B. Panda and W. Perrizo, "Maintaining Surrogate Data for Query Acceleration in Multilevel Secure Database Systems", In Lecture Notes in Computer Science, No. 1006, Information Systems and Data Management, Editor: Subhash Bhalla, Springer-Verlag, November 1995.

[Perr93] W. Perrizo and B. Panda, "Query Acceleration in Multilevel Secure Database Systems", In Proceedings of the 16th National Computer Security Conference, Baltimore, MD, September 1993.

[PCCIP97] "Critical Foundations: Protecting America's Infrastructure", The Report of the President's Commission on Critical Infrastructure Protection, Washington, DC, October 1997.

ARCHITECTURES FOR KNOWLEDGE BASES

Jerry Potter
Professor
Department of Mathematics and Computer Science

Kent State University
Kent, OH 44242

Final Report for:
Summer Faculty Research Program
Rome Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Research Site
IFTB

August 1998

ARCHITECTURES FOR KNOWLEDGE BASES

Jerry Potter
Department of Mathematics and Computer Science
Kent State University
Kent, Ohio 44242
potter@mcs.kent.edu

Abstract

Potential challenge problems were analyzed to determine the nature of their knowledge/database components. The solution to many of these problems depends on an advanced data/knowledge base capability. However, research in not needed on how to build larger data/knowledge bases. Research is needed on how to make knowledge in knowledge bases useful, how to get it to the user over long distances and from multiple sources, how to discover knowledge in unstructured knowledge bases, how to use semi-structured knowledge, how to access data using "conventional communication skills", how to "integrate" knowledge from diverse sources and how to keep knowledge secure. Several current technologies were reviewed to determine their appropriateness for addressing these issues. It was determined that associative techniques combined with processor in memory architecture is the most promising technology.

ARCHITECTURES FOR KNOWLEDGE BASES

Jerry Potter

1.0 INTRODUCTION

Some of the potential challenging problems facing the Air Force today deal with defense information warfare, automated information processing from sensor to shooter, global access to intelligence information, the ability of expeditionary forces to pick up and go any where at any time, the prevention of terrorism through knowledge discovery, using knowledge to provide the ability to react to biological warfare threats and real time planning [II98][1]. One common aspect of all these problems is that they rely heavily on data, information and knowledge. With today's sophisticated data gathering technologies, more data is being collected than ever before. However, it is impossible to extract information from all the data gathered in a timely fashion. The result is a complex collection of knowledge, information, structured and unstructured data. Methods must be found to sort through all of this data and organize it so that the person facing the problem who makes the decisions, whether a database expert or not, whether in an office, on a plane, or in the field has access to the information they need to do their job. The purpose of this study to identify the best possible approaches to discovering, accessing and supplying the diverse and remote knowledge and information needed to solve today's and tomorrow's challenging situations.

The rest of this report is organized as follows. The next section is a brief review of data and knowledge bases, knowledge discovery, parallelism and semiconductors. The third section briefly identifies some technology shortfalls and describes a potential solution to delivering information to those who need it. The fourth section discusses methods to address the

shortfalls in more detail. The fifth section is the conclusion. References follow.

2.0 BACKGROUND

The advent of ever faster computers has created problems as well as solving them. In particular, more data is collected today than ever before. There are two problems. One is the overwhelming amount of data. There is more data in data and knowledge bases than can possibly be understood. There are orders of magnitude more "raw" data waiting to be processed. Second, the data and knowledge is not necessarily easy to access. Even given today's world wide access to networks, remote field terminals may have restricted capacity and limited computing power.

Modern database technology has attacked the abundance of data using various techniques such as active agents and "push" technology whereby data is processed in the background and relevant information is "pushed" to your attention. "Pull" techniques go after the data requested by the user. Databases have evolved into knowledge bases where active components can filter and search data with the efficiency of a librarian. Yet these techniques are not enough. Knowledge discovery in unstructured data in the form of newspapers, telephone conversations, TV, photos, raw electronic transmissions, etc. is most likely to be the source of the most current information.

Research into knowledge discovery has been on going for many years. Automatic program understanding is a paradigm for "data/knowledge" discovery in that programs are known to contain knowledge, yet they can be very difficult to understand. "Automated recognition of abstract concepts is ... difficult to achieve. ... Thus, ..., partial recognition [is

needed]."[Kozaczynski, 1992, p. 217].[2] Typical approaches use syntactic knowledge of the domain to build a model of the abstract concepts to be recognized, starting with a small model and evolving it as the domain becomes better understood.

The recognition process requires syntax and semantics. For example, in programming, the "read data, process data" sequence is almost universal. The fact that the data must be read before it is processed can be specified as a constraint. Facts and relationships such as this can be incorporated into knowledge based rules.

Parallel computing was extensively explored in early years as a solution to the computing needs of intractable problems. The hypothesis was that at some point, large scale integration would reach an upper bound beyond which it would be physically impossible to improve and then parallelism would take over to allow large numbers of chips to be grouped together into the needed super computers. However, chip capacities have continued to improve. As individual computers became more powerful and operating systems more sophisticated, it was felt that explicit parallel machines were not necessary except for special applications and, that networks of computers could be used as large parallel machines for most other applications.

However, the problem today is that large scale integration is so successful that engineers are having difficulty determining what the most effective use of the available real estate is. [Brockman, 1997] says that "classical architectures are already reaching their limits ...[p. 2]".[3] One of the critical barriers to improved performance of conventional architectures is the "gap between the needs of a modern CPU core and the capabilities of the memory system." [Keeton, 1997, p.2][4] - the I/O bus bottleneck. "An alternative approach to system organization,

that promises to deliver high performance with the greatest silicon efficiency, is to place the processing logic on the same die as high density memory, especially DRAM [Brockman, 1997, p. 3]." Parallelism is needed to make the most efficient use of chip real estate in the Processor in Memory (PIM) approach.

3.0 A SMART KNOWLEDGE BASE ACCESSING MACHINE

Figure 1, "Knowledge Based Technology for AFRL Vision"[II98], from the 1998 Information Institute Minnowbrook conference identifies several technology shortfalls based on the potential challenge problems listed in Figure 2 [II98]. In the Processor category, Very Large Data/Knowledge Bases, Parallelization, Contextual Understanding, Special Purpose Architectures and Smart Chips were listed as areas of technology shortfall. The technology shortfalls in the intelligent software include data capture, formal methods, knowledge discovery, learning, information maintenance, mobile computing, mediation, inference and ontology. Considerably synergy can be obtained if the shortfalls from the intelligent software and processor categories are addressed together.

For example, many of the items from both categories such as very large data/knowledge bases, parallelization, mobile computing, formal methods, learning, knowledge discovery, ontology, contextual understanding and smart chips are addressed in the development of a smart portable field deployable terminal (a Smart Knowledge Base Access Machine - SKBAM) and continent software. Since the challenge problems are not mass market issues, it is unlikely that COTS equipment, hardware or software, will be developed to address the shortfalls in a manner consistent with the challenge problems.

Knowledge Base Technology for AFRL Vision

	Intelligent Software	Processors	Mass Storage
Global Awareness	High Performance Knowledge Base Tech Agent Technology **Data Capture** Intelligent Extraction **Formal Methods** Information Access and Retrieval **Knowledge Discovery** Information Fusion **Learning** Data Translation Intelligent Interfaces Modeling and Simulation **Information Maintenance** Heterogeneous Databases Multi-modal	Image Processing **Very Large Data/Knowledge Bases** Data Reduction Signal Processing Voice Text Real Time Size Weight & Power **Parallelization** **Contextual Understanding**	Optical Storage **Intelligent Mass Storage** Terabyte Storage **Storage Organization**
Global Information Exchange	Distributed Processing Network Management Collaboration **Mobile Computing**	Bandwidth Memory **Special Purpose Architectures** **Connectivity (smart)** MEMS	Rapid Transfer Rates Fiber Optics Routing
Dynamic Planning & Execution	Scheduling Planning Optimization Routing Operations Research Artificial Intelligence Decision Aids **Mediation** **Inference** **Ontology**	Hardware Configurability **Smart Chips**	**Content Addressability** Caching **Aggregation**

Bold Face: Technology Shortfalls

Figure 1 - Knowledge Base Technology for AFRL Vision

Potential Challenge Problems

- Defense Information Warfare
- Sensor To Shooter
- Access to Intelligence Information (Any Where - Any Time)
- Expeditionary Force (Pick Up and Go)
- Knowledge Discovery for Terrorism Prevention
- Information Countermeasures For Biological Warfare
- Real Time Planning

Figure 2 - Potential Challenge Problems

It is anticipated that the application problems would require the following scenario for a SKBAM:

1) The field deployed user will verbally query a remote knowledge base of databases. Simple queries will be answered directly. More complex queries will be referred to the appropriate knowledge bases and or raw data.

2) Pertinent data from the specified sources will be anticipated and pre-loaded to the hand held device in a local data/knowledge base (DKB) suitable for interactive use.

3) The user will query the DKB on the device interactively. As needed, new data from previously identified and new sources will be inter-actively sought, down loaded and integrated into the local DKB.

Consider for example, dealing with knowledge discovery in unstructured data. By definition, the data is unstructured so that it is impossible to organize it in a form suitable for conventional database operation. This problem is addressed by an interactive man/machine interface to the DKB. The user would ask or suggest what the system should look for. Based on what was found, the user would make new suggestions. This is a combination of push/pull technology. The pull part limits the processing to what the user of the SKBAM is immediately interested in. The push part informs the user what was found to be of interest based on his queries (and a knowledge base of past queries) even though the user did not specifically request the information. Figure 3 below gives some of the attributes of such a device, capable of accessing very large, distributed, incomplete, structured and unstructured knowledge and databases for knowledge discovery and retrieval at a remote location.

Smart
Teachable
Fast
Portable
Small
Low cost
Low power
Mobile
Natural language
Associative
Figure 3 - SKBAM Attributes

A SKBAM would need application/OS software that would allow its local smarts to i) interact with global/remote smarts, ii) "pre-load" its memory with anticipated background knowledge bases, iii) learn what kind of information is normally requested to help formulate queries, iv) learn to retain down loaded information so it does not have to be down loaded more than once, and v) most importantly, application software to allow it to work with very large assemblages of incomplete DKB and raw data to perform such tasks as knowledge discovery and integration.

Figure 4 is a symbolic functional illustration of a SKBAM in the field. The user is in the field surrounded by intelligent tools which are capable of accessing a vast variety of local information and data and knowledge bases which are linked to the outside world via a communication link which no matter how big is small compared to the amount of data in the outside sourses.

Smart, voice recognition enabled SKBAMs are possible. A chip "... could include sufficient computing power to enable speaker trained, isolated-word speech..." A chip " ...with sufficient performance and 4 to 16 MB of memory to hold the dictionary, when combined with

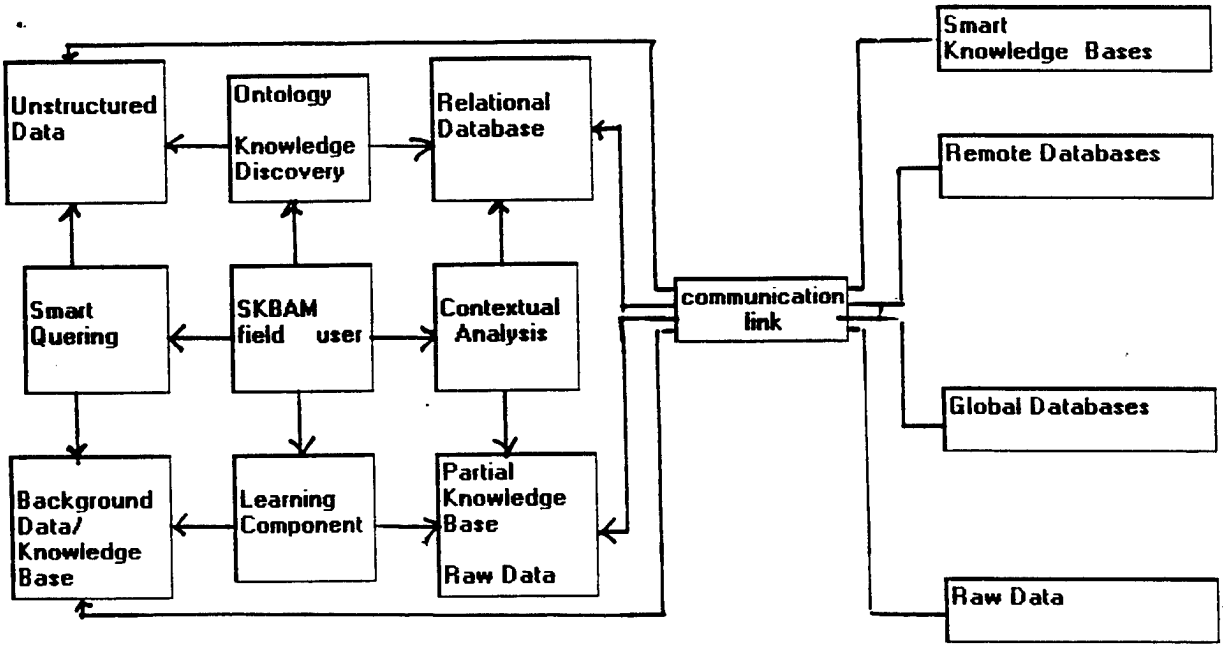


Figure 4 - SKBAM in the Field

the advantages of energy efficiency and small board area, could be an attractive building block for the next generation of PDAs [Patterson, 1997]."[5]

4.0 ASSOCIATIVE COMPUTING

This section reviews some of the aspects of associative computing that are appropriate for a SKBAM environment. First is associaitvie data processing, it can serve as an adjunct to relational database technology because it can handle real time dynamic structured, semi-structured and unstructured data. Associative languages can efficiently implement the real time dynamic rule processing needed for knowledge bases. Second, associative structure codes and unification are well suited for knowledge discovery. Third, a PIM implementation of a multiple associative processor would provide the compact computing power needed for remote, portable comuters.

4.1 Associative Data Processing

The relational database technology and the knowledge base techniques of today are outstanding. But conventional knowledge base technology is too structured. It is unlikely that the "syntax and semantics" of a specific terrorist activity such as that in east Africa can be specified in detail in advance. Unfortunately, the current research trend into object oriented databases is heading in the wrong direction for SKBAM application since "Object data models are very rigid and inflexible ...[Burleson, 1998, p.3]"[6] A more interactive, informal, associative approach is needed.

Associative search and databases have been researched and discussed for many years, and eventhough they are still primitive compared to relational techniques, they are ideal for the remote, real time interactive processing environment of SKBAM. Associative searching is a depth first search process as opposed to the relational database breadth first. As a result, associative searching is faster in SKBAM situations where a first answer is important.

Many rule based languages use an associative style of programming in that the patterns are designed to find data records based on partial information, thus identifying the unknown information in the record by association. Associative capabilities are present in the "alist" facilities of Lisp and when used are found to be a very effective, powerful technique, but they were never fully developed and are primitive and somewhat limited and slow.

Several languages, such as ASC[7][Potter, 1992], have been designed based on associative principles. The main advantage of these techniques over conventional database languages is that they are not only capable of processing unsorted database data, but also "unstructured" data such as raw text and imagery. Associative languages have a natural

language like syntax, not a mathematical set based syntax like SQL and thus are easier for non-database specialists to learn and use and they are suitable for voice input.

Current formal methods such as the associative calculus provide a frame work for real time learning of ontological extensions for the various challenge problems. However, additional formal methods are needed for expanding the capability of associative techniques and languages.

4.2 Structure Codes and Unification

Other associative techniques such as structure codes and associative unification can be very useful when dealing with knowledge discovery and large amounts of unstructured data.

A structure code is a tag containing structural information which is attached to each datum. The advantage of structure codes is that data does not need to be physically reorganized. The datum is static in memory, structure is added by modifying the code as processing progresses. Structure codes can be used to learn at the software and hardware levels. For example, at the hardware level, a command may state that all data with codes that match pattern "12x3y" connect to the "yy" bus. Commands like this can "reconfigures" the busses to fit the current processing needs.

Knowledge discovery requires contextual understanding. The task consists of looking for context in a sea of "unstructured" data. Needless to say, it is a very complex and time consuming task, one requiring pattern matching and unification. Conventional unification is very structured. It allows variables and constants in the data and patterns, but the syntax must be matched (possibly recursively) exactly. In an associative pattern, field names are variables to be bound, values are constants to be matched in the data. Matched data items have values

and structures associated with (bound to) them. In associative unification, knowledge is discovered by extracting the values "associated" with the matched data.

Efficient processing of very large data and knowledge bases requires an efficient implementation of the low level functions. Some languages such as ASC have special control structures to facilitate context sensitive analysis. ASC contains, ANDIF and ANDFOR statements that allow sequences of context sensitive searches to be chained together. Only if the entire chain (or subchain) is successfully matched does the specified action take place. Each individual ANDIF or ANDFOR statement in the chain searches for data in a context free manner. The chain itself specifies the contextual constraints required for a successful search. The ANDFOR and ANDIF constructs depend heavily on efficient low level data parallel hardware.

4.3 Associative Hardware

Computing power is often taken for granted today, however, it is a critical issue for many of the challenge problems for two reasons. First is the sheer volume of data that must be processed. Second is the fact that remote field deployable devices will be limited by size and power. Parallelization will be an important issue in maximizing computing power in deployable devices.

The main problem of computer design today is that the von Neuman computer suffers from the 32 bit memory to CPU bottle neck. Memories can deliver data at high rates and the CPUs can process data at high rates, but moving the data from the memory to the CPU has proven difficult. DARPA has recently begun supporting research into combining CPU on the memory chips. This Processor in Memory (PIM) or Intelligent RAM (IRAM) approach helps

to solve the von Neuman bottleneck. Since both components are on the same chip, memory to CPU busses can be as wide as needed. Their width is not limited by chip pin restrictions.

One avenue that has been explored to use the wider memory-to-cpu busses is to fetch 2,4 or 8 instructions (or 64, 128, or 256 bits of data, respectively) at one time. These "wide word" (or Very Long Instruction Word - VLIW) designs, however, suffer enormously from the basic programming limitations placed on them by the von Neuman model. That is, in standard sequential programming, processing varied data is accomplished by branching around the exceptions. In normal operations, branches are executed every 3 or 4 instructions. A branch is a discontinuity in program flow. This was not a problem in the basic design of 30 years ago when each instruction was essentially discrete. But today's architectures have been modified with layers of cache memories and pipelines to work best when instructions and data flow continuously and smoothly. The wide word technology depends on software - compilers and profilers to "fix" the branching problem. This is a poor solution, the best software can not make up for a poor hardware design. Alternative PIM chip designs, supported by DARPA, promise to facilitate the solution of many difficult problems, but they may not be the best for the very large knowledge base processing demands of Global Awareness and other challenge problems.

PIM is an important technology for knowledge base problems because it can use on chip integration and parallelism to address the von Neuman (I/O bus) bottleneck and special designs can be made to facilitate data/knowledge base processing , allowing data to be found, accessed and processed faster and with less power. That is, since searching very large amounts of data is a fundamental operation in SKBAM applications, the more data that can be loaded and processed at a time the better. Conventional von Neumann designs require approximately

five 32 bit words to be accessed from memory for every 32 bits of data processed. Data parallel designs can process up to 2,000 bits of data (or more) with the same amount of effort.

Special purpose hardware can substantially speed up the search process. Since the cost of a chip is basically constant depending mostly on the volume produced, this hardware can be included on a PIM design at little extra cost. The cost of a device is a function of the chip cost but also is greatly influenced by the cost of connecting the chips together. If the equivalent special purpose hardware were put on a separate chip so that COTS memory and CPUs could be used, this design not only incurs the cost of the special chip, but the additional cost of connecting the chips together and the cost of powering all of the chips, so that even if the COTS chips are very cheap, the total cost of the COTS plus special chip unit is likely to be as much if not more than the total cost of a PIM unit. This is of course an over simplification, many factors are important in determining unit cost. This discussion is only intended to illustrate why a complete PIM design with special purpose hardware can be cheaper on a cost basis than a COTS based design and should be seriously considered.

[Brockman, 1997] claims that PIM is the only way to overcome the I/O bus bottleneck and discusses the spectrum of PIM designs. [Bowman, 1997] concludes "Existing architectures, whether simple, superscalar or out-of-order, are unable to exploit IRAM's increased memory bandwidth and decreased memory latency to achieve significant performance benefits [abstract]."[8] [Patterson, 1997] considers four architecture alternatives: SIMD, VLIW, MIMD and vector for PIM. He states "SIMD is a good match to the IRAM technology ..[p. 2]." But he dismisses SIMD for non-technical reasons, such as "... it has received little compiler development ...[p.2]" He decides to study vector architectures for PIM

implementation even though the need for vector pipes limits the amount of parallelism and the scalability of the design. Vector processors also offer limited datum level multi-tasking support.

[Patterson, 1997] overlooks the rich history of SIMD based associative computers such as the STARAN developed at RADC for database and other applications. One of the primary lessons of data/knowledge base research is that the sophisticated analysis in knowledge acquisition requires an iterative process which translates into a feedback loop in the hardware. Vector machines are not well suited to this type of computation.

The results of the RADC research were not moved to commercial applications for other than technical reasons - the QWERTY factor and poor marketing. As Patterson exemplifies, the architecture is still poorly understood. The superiority of such architectures to real-time interactive database applications such as air traffic control is demonstrated by the Navy's ASPRO E2C processor which is a 1980's mil spec $500k version of the STARAN capable of handling almost 2,000 aircraft simultaneously. The FAA has not been able to replicate the effectiveness of this primitive computer (by today's standards) with any other design after over 10 years and 5 Billion dollars of effort.

The main complaint against associative data parallel (ADP) computers has been 1) a low processor utilization rate, and 2) a lack of efficient multi-task support. Current research into multi-instruction stream ADPs addresses both limitations and are well suited to a PIM design. Given a multiple instruction stream APD databases can be partitioned dynamically at run time into mutually exclusive subsets (e.g. based on company name) and then each subset processed in parallel. Counting the number of product types purchased by each company for

example. Then, the database can be repartitioned into mutually exclusive subsets based on product type and processed in parallel again. Counting the number of companies purchasing each product type. Note that the data in the database do not need to be sorted or duplicated or reorganized or contiguous. This is impossible to do using relational database technology.

Figure 5 is a top level logical PIM design. It is not to scale and does not address important semiconductor issues. It is intended to illustrate that a balanced, symmetrical design is possible. This design shows how the following items could be brought together: 1) multiple instruction streams, 2) scalability, 3) associative data parallelism, 4) an efficient shared memory communication system, 5) wide memory to processor data paths, and 6) high bandwidth I/O paths. Of all possible PIM designs, ADP computers offer the best approach to using parallelism for very large data and knowledge bases, contextual understanding, ontological extensions, knowledge discovery and mobile computing because of its ability to use the very wide memory to CPU busses available on chip and 2) their ability to dynamically re-allocate processors among multiple instruction streams.

5.0 CONCLUSION

The solution to many future problems depends on an advanced data/knowledge base capability. However, we do not need research on how to build large data/knowledge bases, we need new research on how to make knowledge in knowledge base useful, how to get it to the user over long distances and from multiple sources, how to discover knowledge in unstructured knowledge bases, how to use semi-structured knowledge, how to access data using "conventional communication skills", how to "integrate" knowledge from diverse sources and how to keep knowledge secure.

Data on a SKBAM is likely to be fragments of knowledge bases, raw data, complete and incomplete data bases and the user will need all of the above features to use it effectively.

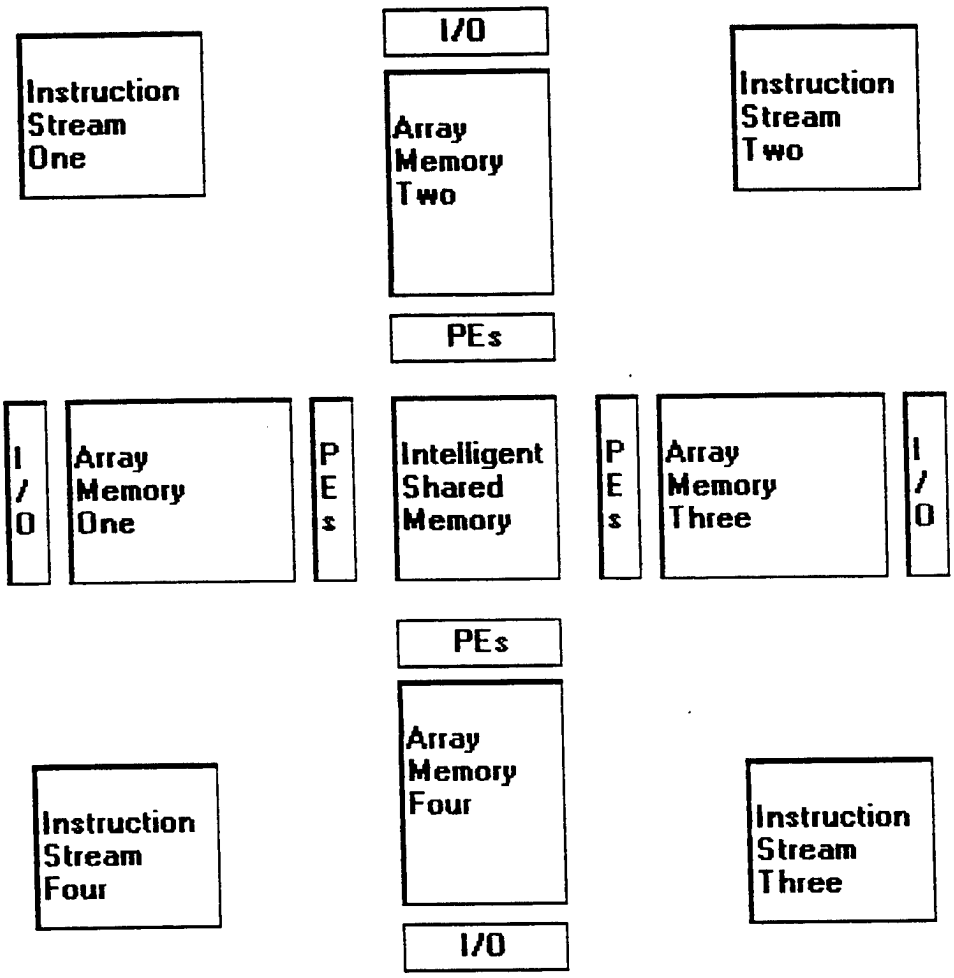


Figure 5 - Top Level PIM Design for a SKBAM

6.0 REFERENCES[910]

1. [II98] Preliminary report of the Intelligent Knowledge Base Systems Working Group of the Information Institute's Workshop 98, June 24-26, 1998, Minnowbrook , NY.

2. [Kozaczynski, 1992] Kozaczynski, Wojtek, Jim Ning and Tom Sarver, "Program Concept Recognition," in Proceedings KBSE '92, McLean, VA., Sept. 20-23, 1992, pp. 216-225.

3. [Brockman, 1997] Brockman, Jay B. and Peter M. Kogge, The Case for Processing-in-Memory, University of Notre Dame, Department of Computer Science and Engineering, TR-97-3.

4. [Keeton, 1997] Keeton, Kimberly, Remzi Arpaci-Dusseau, and David A. Patterson, "IRAM and SmartSIMM: Overcoming the I/O Bus Bottleneck", University of California at Berkeley, Computer Science Division, http://iram.cs.berkeley.edu

5. [Patterson, 1997] Patterson, D. et. al., Intelligent RAM (IRAM): the Industrial Setting, Applications, and Architectures, in ICCD '97 International Conference on Computer Design, Austin, Texas, October 1997, pp. 10-12.

6.[Burleson, 1998] Burleson, D. K. Inside the Database Object Model, CRC Press, Boca Raton, FL., 1998.

7. [Potter, 1992] Potter, J. L., Associate Computing, Plenum Publishing, New York, 1992.

8.[Bowman, 1997] Bowman, N., N. Cardwell, C. Kozyrakis, C. Romer and H. Wang, "Evaluation of Existing Architectures in IRAM Systems", University of California at Berkeley, Computer Science Division, http://iram.cs.berkeley.edu

9.Carter, J. W. Hsieh, L. Stoller, M. Swanson, L. Zhang, E. Brunvand, A. Davis, Cc. Kuo, R. Kuramkote, M. Parker, L. Schaelicke and T. Tateyama, Impulse: An Adaptable Memory System, retrac@cs.utah.edu.

10.Lipman, A and W. Yang, The Smart Access Memory: An Intelligent RAM for Nearest Neighbor Database Searching.

MODELING AND IMPLEMENTATION OF LOW DATA RATE MODEM USING MATLAB

Salahuddin Qazi
Associate Professor
School Of Information Systems and Engineering Technology

State University of New York Institute of Technology
P.O. Box 3050, Utica, NY 13504-3050

Final Report for:
Summer Faculty Research Program

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC.

and

Air Force Laboratory, Rome Site

September 1998

MODELING AND IMPLEMENTATION OF LOW DATA RATE MODEM USING MATLAB

Salahuddin Qazi
Associate Professor
School of Information Systems and Engineering Technology
State University Of New York Institute of Technology

Abstract

The report investigates methods of modeling and DSP implementation of the modem used for low data rate multi-media communications over wireless links developed and built at the Air Force Lab, Rome site. Communication Toolbox with MATLAB version of 5.1.0.421 was used in Simulink environment to model the modem based communication system. C-code of the simulated system was generated for targeting the DSP board using Real-time workshop (RTW) of the MATH WORKS Inc. Third party software based on MATLAB for the implementation of DSP was also investigated.

INTRODUCTION

RF-5710 HF modem is a part of a prototype Briefcase Communications Terminal developed and built at the Air Force Research Lab, Rome Site, New York. Briefcase communication terminal makes use of radio or antenna system already installed on the aircraft and is used for transmitting clear voice, clear data, secure voice, secure data as well as limited Automatic Link Establishment (ALE) from the aircraft. It can also be used for land, mobile or sea applications. Applications for data include E-mail, ITV (in transit visibility) message, GPS and weather data etc. In addition to the modem, Briefcase communication terminal also contains a notebook computer, an encryption unit, a keyline relay, audio VOX for ALE and a 24- volt power supply.

Briefcase communication terminal is intended to be a simple carry on system adaptable to several aircraft without any modification to the aircraft. The current size of the modem makes the system inconvenient to hand carry it. The purpose of the project is to investigate an efficient way of modeling the modem and implementing it on a DSP chip located on the note book computer using suitable software. The output of the modem is required to be an audio signal with a bandwidth of up to 3400 Hz. This will eliminate the need for a hardware modem making the low data rate communication system lighter and easy to transport. Since the DSP chip is configurable, it can be used for programming of variety of communication applications.

There is an increased trend in communications technology to move from analog to digital, where increasingly more functions of the current radio systems are implemented in software leading toward the software radio. The term software radio was coined by Joseph Mitola III in 1991 and is defined as radio whose channel modulation waveforms are defined in software. In the radio transmitter waveforms are generated as sampled digital signals, converted from digital to analog via a wide band digital to analog converter and then possibly up converted from IF to RF. The radio receiver employs a wide band analog to digital converter that captures all of the channels of the software radio node. The use of software radios helps to free hard ware based requirement, making radios more compatible to standards, and allows to adapt with technological changes. DSP chips will allow engineers to produce hardware/software solution where the hardware continues to be used and software gets upgraded [1].

In the USA, the government started a Speakeasy Multi-band Multi-mode Radio (MBMMR) program at Rome Laboratory, Rome, New York, for next generation Military radio . It is a combination of various strategies including waveform reconfigurability in modular and programmable signal processors through software implementation of waveforms and associated processing [2]. In the U.K. the Bowman program was started which required the radio elements to have the same external interfaces and operational look and feel. Speakeasy program outgrew into Modular Multifunction Information Transfer System (MMITS) Forum which created "PC of the radio world" using open architecture. It attempts to foster the ability to build every type of communications device from a common set of pieces. The Communication Division of Harris Corporation has developed tactical radios called Falcon II family from a common set of electrical,

mechanical and software elements based on quasi -open architecture [3].

Methodology

The aim of our project was to replace the RF-5710 HF modem with a DSP chip or board and make it adaptable to changing technologies. This required modeling and simulation of the modem and generating a C-code to implement it on a DSP chip. A number of communication software packages including ACOLADE (Advanced Communication Link Analysis and Design Environment) and System View by ElANIX were reviewed. It was decided to use MATLAB and its extension Simulink to simulate a communication system (modem) because of its universality and third party support available for DSP implementation in a real-time environment. Real-time workshop is used to generate C-code and implement on a DSP chip using MATLAB supported software tools. It is also desired to implement part of modem on the notebook (Pentium) processor and part on the programmable DSP chip in real time because of the low data rate. Control of frequencies may be implemented on the central processor by way of partitioning it.

Based on MATLAB and Simulink, Real-time workshop (RTW) provides a common development environment and rapid prototyping of real-time DSP software. This is obtained by SPOXWorks, which is an add-on to the Math Works Real-time Workshop that provides a real time environment for running models produced with the Simulink diagram tools. Real-time workshop generates ANSI C-code for Spectron SPOXWorks targets, including Loughborough LSI PC32 and other DSP boards. SPOXWorks based DSP applications can be developed under DOS and windows on PCs, and under Unix on workstations. The target environment can be an attached workstation processor, a homogeneous embedded system, a heterogeneous Embedded system, or a Pentium processor in a native PC environment. In a Native PC environment, SPOXWorks runs directly on the Pentium processor host as a VxD at windows' Ring 0. In this environment, SPOXWorks augments Windows with a predictable real-time environment that is ideal for executing multimedia functions like wave-table audio synthesis, together with data acquisition and test and measurement. SPOXWorks is currently available for the following targets. It currently has three platforms and can be targeted on few DSP chips including PCMCIA card.

Texas Instruments	Motorola	Analog devices	NEC	Intel
TMS320C3x	56156/166	2106x(SHARC)	upd 77016	Pentium
TM320C4x	5600x		upd 77017	Processor
TM320C5x	563xx			
TM320C8x(MVP)				

Other MATLAB supported DSP development systems are also reviewed. Mercury computer system offers a RACE MATLAB Math Library which enables M-file programs developed in the Math works to be targeted to Race i860 (Intel)and power PC embedded computer system. This eliminates the need to manually convert M-files to C-code. Once a MATLAB application is implemented on the RACE system , life cycle support and

functional evolution is greatly facilitated. By using MATLAB as a "common language" real time system can save significant time. dSPACE GmbH of Germany also offers a total development environment, providing unmatched flexibility and processing power for rapid control prototyping and hardware-in-the-loop simulation in real time. This environment provides a seamless design flow for real-time systems. It utilizes MATLAB for system identification and modeling. Simulink is used for off-line simulation and real-time Workshop for generation of C-code. All I/O specifications as well as interrupts can be defined in the Simulink environment by making use of dSPACE's Real-time Interface software which automatically insures the correct interfacing of the generated C-code. Real-time hardware include DSPs like Alpha AXP 21164 (1000 MFlops) and TMS320C40/TMS320C31 (60 M Flops).

Results

The modem to be replaced by the software is half duplex, serial (single tone) modem compatible with MIL-STD-188-110A and is used for all line of site modes including HF, VHF and UHF. It is capable of transmitting data at a rate of 75 to 4800 bps and is typically used at 2400 bps for good error correction. The output of the modem is required to be an audio signal with a bandwidth of up to 3400 Hz.

Two models were developed to test the simulations and generate the C-code. The first model consists of a simple transmitter having a QPSK modulator with an input of 2400 bps and a similar demodulator in the receiver. The purpose of this model was to test the capability of Math Works DSP Workshop components consisting of simulink, Real-time Workshop, DSP Blockset and various tool boxes which were loaned from MathWorks, Inc. The second model shown in figure 1, contains binary sequence convolution encoder, a buffer, vector to scaler converter and a terminator in addition to MPSK modulator and a square wave signal generator as an input source. The receiver consists of MPSK demodulator, scalar to vector converter, unbuffer, binary sequence convolution decoder, a terminator, oscilloscopes and an eye-diagram scatter plot to study the channel performance and inter-symbol interference. Rayleigh noise is also introduced to study multipath effect in a HF channel. C- code was generated for the second model.

For the first model a square wave generator of 2500 bit/sec with a unipolar pulse of +1 volt is used as an input waveform. The QPSK modulator and demodulator were set to the following parameters with two different carrier frequencies.

Symbol interval = 0.0005 sec.
Sample time = 0.00005 sec.
Initial phase (rad)=0
Carrier frequencies =5000/2/pi and 10000/2/pi

Oscilloscope traces of input and output waveforms for the first model at two different carrier frequencies are shown in figures 2(a), 2(b), 2(c) and figures 3(a), 3(b) and 3(c). C-code of the second model is obtained by MATLAB's Real- time Workshop (RTW) and is shown in the appendix.

Conclusion

MATLAB in the simulink environment was used to model and simulate the radio modem for the Briefcase communication terminal. C-code for the simulated modem was generated using MATLAB Real-time workshop (RTW). Investigation on the use of MATLAB third party software showed that SPOXWorks can be easily used with Simulink/RTW for the real time implementation of some DSP boards including PCMCIA/PC cards. The work on the real time implementation was not completed due to the lack of time and funds. Results of the simulation showed that MATLAB in the Simulink environment is flexible, affordable and can be used to implement radio modems which will make the Briefcase communication terminal portable and adaptable to changing technologies.

Further work is needed to implement the C-code on the lap top processor for real time applications and the possibility of partitioning the tasks by using additional DSP for modem functions. Field programmable gate arrays (FPGAs) is an important alternative for DSP implementation and is very powerful and reconfigurable. Since low power is an important criteria for the lap top applications, it will be an important study to examine the power consumption, the size, weight and flexibility tradeoffs of the DSP versus the pure FPGAs versus a mix of a DSP and an FPGAs on the same chip.

Acknowledgements

The author would like to thank the Air Force Office of Scientific Research, Bolling AFB Washington, DC, and the Air Force Lab, Rome site, for enabling the project Under the Summer Research Faculty Program. Special thanks are due to Joseph Mancini, Larry Spadaro of the Air Force Laboratory, Rome site, for numerous discussions and help. Discussions via E-mail with Joe Mitola of the MITRE Corporation, Mclean, VA, have also been very helpful and his help is greatly appreciated.

References

1. Joe Mitola, "The Software Radio Architecture". IEEE Communications Magazine, PP 26-38, May 1995.
2. Raymond J. Lackey, Donald W. Upmal, "Speakeasy: The Military Software Radio", IEEE Communications Magazine, PP56-61, May 1995.
3. Andy Ivers and Dave Smith, "A Practical Approach to the Implementation of Multiple Radio Configuration Utilizing Reconfigurable Hardware and Software Building Blocks", Military Communications Conference, PP 1327-1332, 1997.

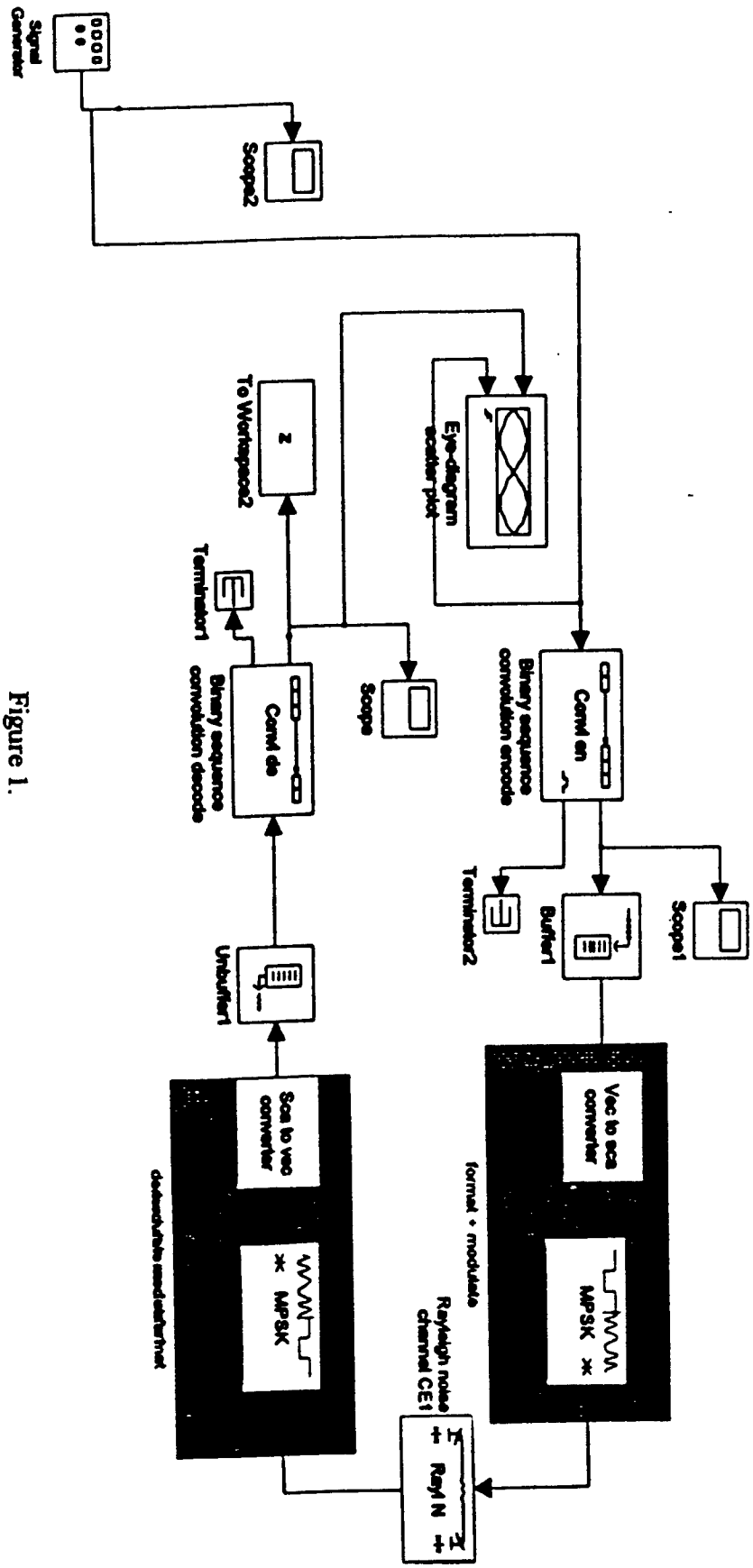

Signal Generator
Scope2
To Workspace2
z
Eye-diagram scatter plot
Terminator1
Binary sequence convolution decode
Conv de
Scope
Binary sequence convolution encode
Conv en
Terminator2
Buffer1
Scope1
Unbuffer1
Sca to vec converter
* MPSK
Vec to sca converter
MPSK *
format + modulate
Rayleigh noise channel CE1
Rayl N

Figure 1.

Fig. 2(a) Input to the QPSK modulator
Waveform = Square
Amplitude = 1 volt
Frequency = 2500 Hz

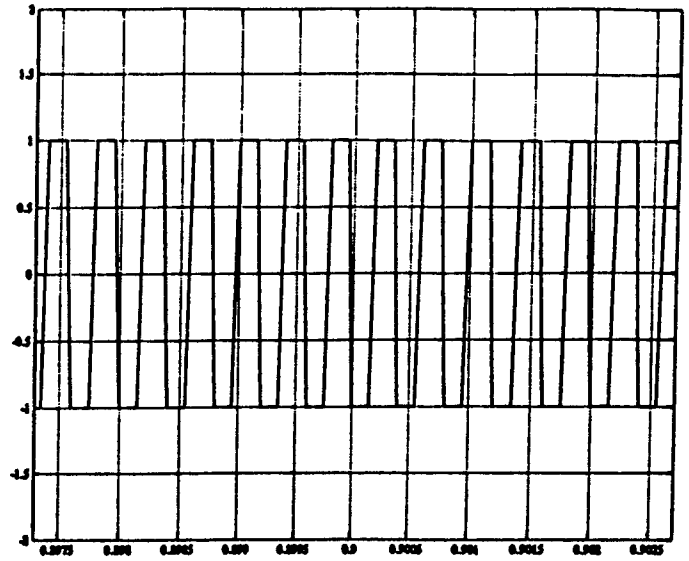


Fig. 2(b) Output of QPSK modulator
Symbol interval = 0.0005 seconds
Sample time = 0.00005 seconds
Initial phase (rad) = 0
Carrier frequency = 5000/2/pi

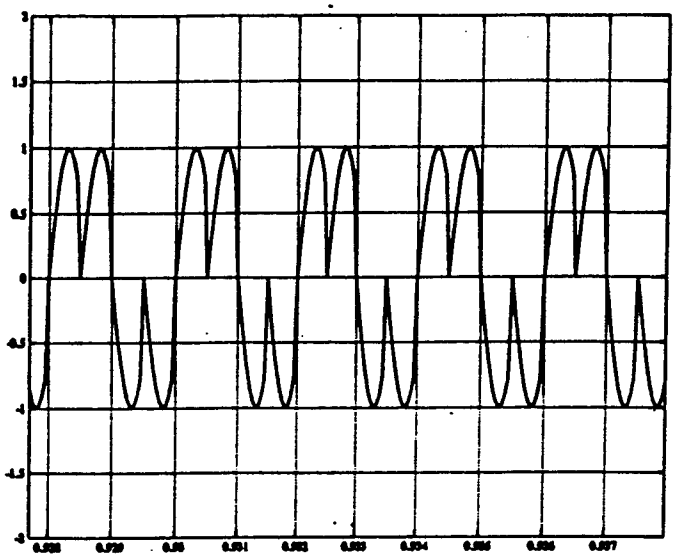


Fig. 2(c) Output of QPSK demodulator
Symbol interval = 0.0005 seconds
Sample time = 0.00005 seconds
Carrier frequency = 5000/2/pi

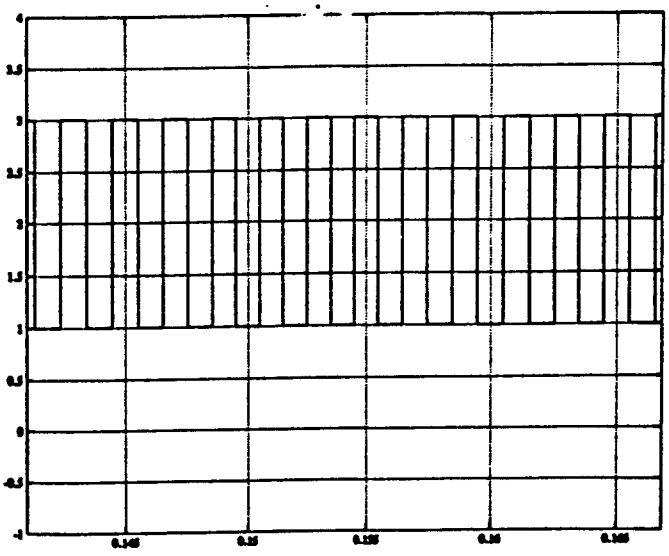


Fig. 3 (a) Input to the QPSK modulator
Waveform = Square
Amplitude = 1 volt
Frequency = 2500 Hz

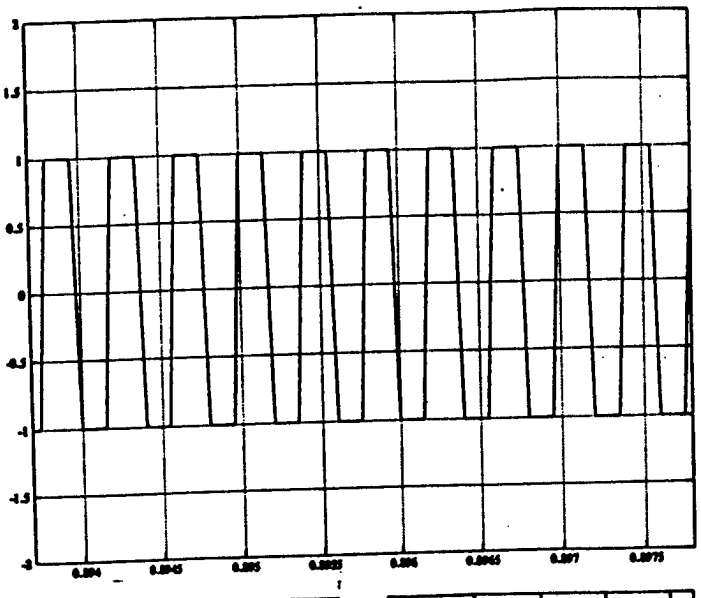


Fig. 3(b) Output of QPSK modulator
Symbol interval = 0.0005 seconds
Sample time = 0.00005 seconds
Initial phase (rad) = 0
Carrier frequency = 10000/2/pi

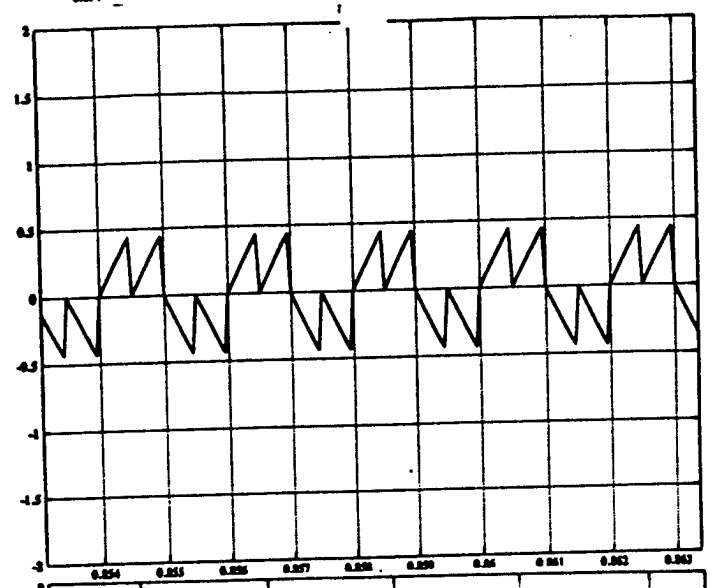
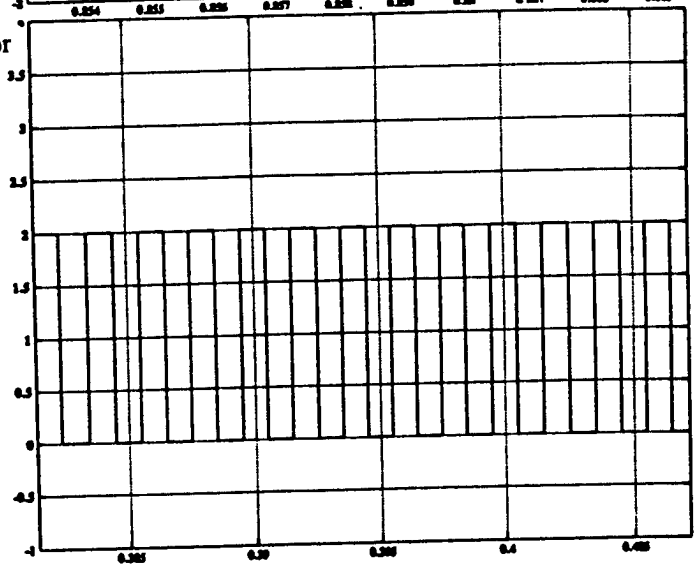


Fig. 3(c) Output of QPSK demodulator
Symbol interval = 0.0005 seconds
Sample time = 0.00005 seconds
Carrier frequency 10000/2/pi



```
sIsSampleHit(rtS, 1, tid)) {
    /* Constant Block: <S33>/one1 */
    rtB.s33_one1 = rtP.s33_one1.Value;
  }

  /* Sample hit for TID=10 */
  if(ssIsSampleHit(rtS, 10, tid)) {
    /* RelationalOperator Block: <S33>/Relational Operator */
    rtB.s33_Relational_Operator = (rtB.root_Discrete_Pulse_Generator2 >= rtB.s3
  }

  /* Sample hit for TID=1 */
  if(ssIsSampleHit(rtS, 1, tid)) {
    /* Memory Block: <S33>/Memory1 */
    rtB.s33_Memory1 = rtRWork.s33_Memory1.PrevU;

    /* RelationalOperator Block: <S33>/Relational Operator1 */
    rtB.s33_Relational_Operator1 = (rtB.s33_Memory1 >= rtB.s33_one1);

    /* RelationalOperator Block: <S33>/Relational Operator2 */
    rtB.s33_Relational_Operator2 = (rtB.s33_Relational_Operator > rtB.s33_Relat

    /* Level1 S-Function Block: <S32>/Modulo operation1 (sviterbi) */
    {
      SimStruct *rts = ssGetSFunction(rtS, 19);
      real_T *sfcnU = ssGetU(rts);
      real_T *sfcnX = ssGetContStates(rts);
      real_T *sfcnY = ssGetY(rts);

      /* copy non-contiguous inputs into s-function work vector */
      {
        real_T *uPtr = sfcnU;
        *uPtr++ = rtB.s9_Integer_scalar_to_vector[0];
        *uPtr++ = rtB.s9_Integer_scalar_to_vector[1];
        *uPtr++ = rtB.s9_Integer_scalar_to_vector[2];
        *uPtr++ = rtB.s33_Relational_Operator2;
      }

      sfcnOutputsLevel1(sfcnY, sfcnX, sfcnU, rts, tid);
    }
  }

  /* Sample hit for TID=11 */
  if(ssIsSampleHit(rtS, 11, tid)) {
    /* S-Function Block: <S19>/ZOH */
    rtB.s19_ZOH[0] = rtB.s32_Modulo_operation1[0];
    rtB.s19_ZOH[1] = rtB.s32_Modulo_operation1[1];
  }

  /* S-Function Block: <S19>/Complete Unbuffer (sunbuffc) */

  {
    if (ssIsSampleHit(rtS, 10, tid)) {
      /* Output the current sample: */
      real_T *buf = &rtRWork.s19_Complete_Unbuffer.circ_buf[0];
      real_T *outBuf = rtPWork.s19_Complete_Unbuffer.OUTBUF_PTR;
      rtB.s19_Complete_Unbuffer = *outBuf++;
      if (outBuf == buf + 4) {
        outBuf = buf;
      }

      rtPWork.s19_Complete_Unbuffer.OUTBUF_PTR = (void *)outBuf;
```

```
  }
  if (ssIsSampleHit(rtS, 11, tid)) {
    /* Store the next input vector: */
    real_T *inBuf = rtPWork.s19_Complete_Unbuffer.INBUF_PTR;

    *inBuf++ = rtB.s19_ZOH[0];
    *inBuf++ = rtB.s19_ZOH[1];

    {
      real_T *buf = &rtRWork.s19_Complete_Unbuffer.circ_buf[0];
      if (inBuf == buf + 4) {
        inBuf = buf;                    /* wrap circular buffer */
      }
    }
    rtPWork.s19_Complete_Unbuffer.INBUF_PTR = (void *)inBuf;
  }
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* Memory Block: <S26>/Memory1 */
  rtB.s26_Memory1 = rtRWork.s26_Memory1.PrevU;
}

/* Level1 S-Function Block: <S22>/S-function1 (homopuls) */
{
  SimStruct *rts = ssGetSFunction(rtS, 20);
  real_T *sfcnU = ssGetU(rts);
  real_T *sfcnX = ssGetContStates(rts);
  real_T *sfcnY = ssGetY(rts);
  sfcnOutputsLevel1(sfcnY, sfcnX, sfcnU, rts, tid);
}

/* Sample hit for TID=4 */
if(ssIsSampleHit(rtS, 4, tid)) {
  /* Level1 S-Function Block: <S23>/S-function (regshift) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 21);
    real_T *sfcnU = ssGetU(rts);
    real_T *sfcnX = ssGetContStates(rts);
    real_T *sfcnY = ssGetY(rts);

    /* copy non-contiguous inputs into s-function work vector */
    {
      real_T *uPtr = sfcnU;
      *uPtr++ = rtB.s20_Complete_Unbuffer;
      *uPtr++ = rtB.s22_S_function1[0];
    }

    sfcnOutputsLevel1(sfcnY, sfcnX, sfcnU, rts, tid);
  }
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* Constant Block: <S26>/one1 */
  rtB.s26_one1 = rtP.s26_one1.Value;
}

/* Sample hit for TID=4 */
if(ssIsSampleHit(rtS, 4, tid)) {
```

```
  /* RelationalOperator Block: <S26>/Relational Operator */
  rtB.s26_Relational_Operator = (rtB.s23_S_function[3] >= rtB.s26_one1);
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* RelationalOperator Block: <S26>/Relational Operator1 */
  rtB.s26_Relational_Operator1 = (rtB.s26_Memory1 >= rtB.s26_one1);

  /* RelationalOperator Block: <S26>/Relational Operator2 */
  rtB.s26_Relational_Operator2 = (rtB.s26_Relational_Operator > rtB.s26_Relat

  /* Level1 S-Function Block: <S25>/Modulo operation1 (sviterbi) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 22);
    real_T *sfcnU = ssGetU(rts);
    real_T *sfcnX = ssGetContStates(rts);
    real_T *sfcnY = ssGetY(rts);

    /* copy non-contiguous inputs into s-function work vector */
    {
      real_T *uPtr = sfcnU;
      *uPtr++ = rtB.s23_S_function[0];
      *uPtr++ = rtB.s23_S_function[1];
      *uPtr++ = rtB.s23_S_function[2];
      *uPtr++ = rtB.s26_Relational_Operator2;
    }

    sfcnOutputsLevel1(sfcnY, sfcnX, sfcnU, rts, tid);
  }

  /* Level1 S-Function Block: <S24>/S-function (regdown) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 23);
    real_T *sfcnU = ssGetU(rts);
    real_T *sfcnX = ssGetContStates(rts);
    real_T *sfcnY = ssGetY(rts);

    /* copy non-contiguous inputs into s-function work vector */
    {
      real_T *uPtr = sfcnU;
      *uPtr++ = rtB.s25_Modulo_operation1[0];
      *uPtr++ = rtB.s25_Modulo_operation1[1];
      *uPtr++ = rtB.s23_S_function[3];
      *uPtr++ = rtB.s22_S_function1[1];
    }

    sfcnOutputsLevel1(sfcnY, sfcnX, sfcnU, rts, tid);
  }
}

/* Sample hit for TID=10 */
if(ssIsSampleHit(rtS, 10, tid)) {
  /* ToWorkspace Block: <Root>/To Workspace */
  {
    const char_T *status;
    real_T *u = &rtB.s19_Complete_Unbuffer;
    if ((status = rt_UpdateLogVar((LogVar*)rtPWork.root_To_Workspace.LoggedDa
      ssSetErrorStatus(rtS,status);
      return;
    }
  }
```

```
  }

  /* Sample hit for TID=1 */
  if(ssIsSampleHit(rtS, 1, tid)) {
    /* ToWorkspace Block: <Root>/To Workspace1 */
    {
      const char_T *status;
      real_T *u = &rtB.root_Relational_Operator;
      if ((status = rt_UpdateLogVar((LogVar*)rtPWork.root_To_Workspace1.LoggedD
        ssSetErrorStatus(rtS,status);
        return;
      }
    }
  }

  /* Sample hit for TID=10 */
  if(ssIsSampleHit(rtS, 10, tid)) {
    /* ToWorkspace Block: <Root>/To Workspace2 */
    {
      const char_T *status;
      real_T *u = &rtB.s24_S_function[0];
      if ((status = rt_UpdateLogVar((LogVar*)rtPWork.root_To_Workspace2.LoggedD
        ssSetErrorStatus(rtS,status);
        return;
      }
    }
  }
}

/* Perform model update */
void MdlUpdate(int_T tid)
{
  /* Level1 S-Function Block: <S28>/S-function1 (homopuls) */
  {
    SimStruct *rts = ssGetSFunction(rtS,4);

    real_T *sfcnU = ssGetU(rts);
    real_T *sfcnX = ssGetContStates(rts);
    sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
  }

  /* Sample hit for TID=1 */
  if(ssIsSampleHit(rtS, 1, tid)) {
    /* Memory Block: <S31>/Memory1 */
    rtRWork.s31_Memory1.PrevU = rtB.s29_S_function[2];

    /* Level1 S-Function Block: <S27>/Modulo operation1 (sconvenc) */
    {
      SimStruct *rts = ssGetSFunction(rtS,6);

      real_T *sfcnU = ssGetU(rts);
      real_T *sfcnX = ssGetContStates(rts);

      /* copy non-contiguous inputs into s-function work vector */
      {
        real_T *uPtr = sfcnU;
        *uPtr++ = rtB.s29_S_function[0];
        *uPtr++ = rtB.s29_S_function[1];
        *uPtr++ = rtB.s31_Relational_Operator2;
      }

      sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
```

```
  }
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* RandomNumber Block: <S49>/Random Number */
  rtRWork.s49_Random_Number.NextOutput[0] = rt_NormalRand((UINT_T *)&rtIWork.
  rtRWork.s49_Random_Number.NextOutput[1] = rt_NormalRand((UINT_T *)&rtIWork.
}

/* Level1 S-Function Block: <S41>/Scheduled reset-int1 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS,9);

  real_T *sfcnU = ssGetU(rts);
  real_T *sfcnX = ssGetContStates(rts);
  sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
}

/* Level1 S-Function Block: <S42>/Scheduled reset-int2 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS,10);

  real_T *sfcnU = ssGetU(rts);
  real_T *sfcnX = ssGetContStates(rts);
  sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* Memory Block: <S34>/Memory1 */
  rtRWork.s34_Memory1.PrevU = rtB.root_Discrete_Pulse_Generator1;

  /* Level1 S-Function Block: <S4>/Modulo operation1 (sconvenc) */
  {
    SimStruct *rts = ssGetSFunction(rtS,13);

    real_T *sfcnU = ssGetU(rts);
    real_T *sfcnX = ssGetContStates(rts);

    /* copy non-contiguous inputs into s-function work vector */
    {
      real_T *uPtr = sfcnU;
      *uPtr++ = rtB.s5_ZOH[0];
      *uPtr++ = rtB.s5_ZOH[1];
      *uPtr++ = rtB.s34_Relational_Operator2;
    }

    sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
  }

  /* RandomNumber Block: <S47>/Random Number */
  rtRWork.s47_Random_Number.NextOutput[0] = rt_NormalRand((UINT_T *)&rtIWork.
  rtRWork.s47_Random_Number.NextOutput[1] = rt_NormalRand((UINT_T *)&rtIWork.
}

/* Level1 S-Function Block: <S37>/Scheduled reset-int1 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS,15);

  real_T *sfcnU = ssGetU(rts);
  real_T *sfcnX = ssGetContStates(rts);
```

```
  sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
}

/* Level1 S-Function Block: <S38>/Scheduled reset-int2 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS,16);

  real_T *sfcnU = ssGetU(rts);
  real_T *sfcnX = ssGetContStates(rts);
  sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* Memory Block: <S33>/Memory1 */
  rtRWork.s33_Memory1.PrevU = rtB.root_Discrete_Pulse_Generator2;

  /* Level1 S-Function Block: <S32>/Modulo operation1 (sviterbi) */
  {
    SimStruct *rts = ssGetSFunction(rtS,19);

    real_T *sfcnU = ssGetU(rts);
    real_T *sfcnX = ssGetContStates(rts);

    /* copy non-contiguous inputs into s-function work vector */
    {
      real_T *uPtr = sfcnU;
      *uPtr++ = rtB.s9_Integer_scalar_to_vector[0];
      *uPtr++ = rtB.s9_Integer_scalar_to_vector[1];
      *uPtr++ = rtB.s9_Integer_scalar_to_vector[2];
      *uPtr++ = rtB.s33_Relational_Operator2;
    }

    sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
  }
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* Memory Block: <S26>/Memory1 */
  rtRWork.s26_Memory1.PrevU = rtB.s23_S_function[3];
}

/* Level1 S-Function Block: <S22>/S-function1 (homopuls) */
{
  SimStruct *rts = ssGetSFunction(rtS,20);

  real_T *sfcnU = ssGetU(rts);
  real_T *sfcnX = ssGetContStates(rts);
  sfcnUpdateLevel1(sfcnX, sfcnU, rts, tid);
}

/* Sample hit for TID=1 */
if(ssIsSampleHit(rtS, 1, tid)) {
  /* Level1 S-Function Block: <S25>/Modulo operation1 (sviterbi) */
  {
    SimStruct *rts = ssGetSFunction(rtS,22);

    real_T *sfcnU = ssGetU(rts);
    real_T *sfcnX = ssGetContStates(rts);

    /* copy non-contiguous inputs into s-function work vector */
```

```
}

/* Level1 S-Function Block: <S12>/Integer vector to scalar (simbi2de) */
{
  SimStruct *rts = ssGetSFunction(rtS, 8);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S41>/Scheduled reset-int1 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS, 9);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S42>/Scheduled reset-int2 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS, 10);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S40>/Maximum (arymimai) */
{
  SimStruct *rts = ssGetSFunction(rtS, 11);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S10>/Integer scalar to vector (simde2bi) */
{
  SimStruct *rts = ssGetSFunction(rtS, 12);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S4>/Modulo operation1 (sconvenc) */
{
  SimStruct *rts = ssGetSFunction(rtS, 13);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S11>/Integer vector to scalar (simbi2de) */
{
  SimStruct *rts = ssGetSFunction(rtS, 14);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S37>/Scheduled reset-int1 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS, 15);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S38>/Scheduled reset-int2 (schdint) */
{
  SimStruct *rts = ssGetSFunction(rtS, 16);
  sfcnTerminate(rts);
}

/* Level1 S-Function Block: <S36>/Maximum (arymimai) */
{
  SimStruct *rts = ssGetSFunction(rtS, 17);
  sfcnTerminate(rts);
}
```

```
  /* Level1 S-Function Block: <S9>/Integer scalar to vector (simde2bi) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 18);
    sfcnTerminate(rts);
  }

  /* Level1 S-Function Block: <S32>/Modulo operation1 (sviterbi) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 19);
    sfcnTerminate(rts);
  }

  /* Level1 S-Function Block: <S22>/S-function1 (homopuls) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 20);
    sfcnTerminate(rts);
  }

  /* Level1 S-Function Block: <S23>/S-function (regshift) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 21);
    sfcnTerminate(rts);
  }

  /* Level1 S-Function Block: <S25>/Modulo operation1 (sviterbi) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 22);
    sfcnTerminate(rts);
  }

  /* Level1 S-Function Block: <S24>/S-function (regdown) */
  {
    SimStruct *rts = ssGetSFunction(rtS, 23);
    sfcnTerminate(rts);
  }
}

/* model registration */
#include "s1.reg"

»
```

AN IMAGE REGISTRATION ALGORITHM BASED ON THE PROJECTIVE TRANSFORMATION MODEL AND AUTOMATED BLOCK MATCHING FEATURE POINT SELECTION WITH APPLICATIONS IN MULTIFRAME INTEGRATION AND CONCEALED WEAPON ENHANCEMENT

Richard R. Schultz
Assistant Professor
Department of Electrical Engineering

University of North Dakota
P.O. Box 7165
Grand Forks, ND 58202-7165

Final Report for:
Summer Faculty Research Program
Rome Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

and

Rome Research Site

August 1998

AN IMAGE REGISTRATION ALGORITHM BASED ON THE PROJECTIVE TRANSFORMATION MODEL AND AUTOMATED BLOCK MATCHING FEATURE POINT SELECTION WITH APPLICATIONS IN MULTIFRAME INTEGRATION AND CONCEALED WEAPON ENHANCEMENT

Richard R. Schultz
Assistant Professor
Department of Electrical Engineering
University of North Dakota

Abstract

A subpixel-resolution image registration algorithm based on the nonlinear projective transformation model is proposed to account for camera translation, rotation, pan, and tilt. Typically, parameter estimation techniques for transformation models require the user to manually select feature point pairs between the images undergoing registration. In this research, the block matching algorithm is used to automatically select correlated feature point pairs between two images, and these features are used to calculate an iterative least squares solution for the nonlinear projective transformation parameters. Since block matching is only capable of estimating accurate translation motion vectors in discontinuous edge regions, inaccurate feature point pairs are statistically eliminated prior to computing the least squares parameter estimate. Convergence of the projective transformation model estimation algorithm is generally achieved in several iterations. Simulations show that the algorithm estimates accurate integer- and subpixel-resolution registration parameters for intensity image sequence frames, dissimilar image slices from the Visible Human Project, and uncalibrated infrared images. Through subpixel-resolution registration, high-resolution video stills are generated by integrating the registered pixels from a short sequence of low-resolution video frames. Experimental results are also shown in utilizing dissimilar data registration followed by vector quantization to segment tissues from multimodality Visible Human Project slices, as well as to enhance concealed weapons imaged by a dual long-wave and short-wave infrared camera.

AN IMAGE REGISTRATION ALGORITHM BASED ON THE PROJECTIVE TRANSFORMATION MODEL AND AUTOMATED BLOCK MATCHING FEATURE POINT SELECTION WITH APPLICATIONS IN MULTIFRAME INTEGRATION AND CONCEALED WEAPON ENHANCEMENT

Richard R. Schultz

Introduction

Many applications within the realm of sensor fusion require the accurate registration of image data. The challenge involves registering not only data acquired from the same sensor array after a transformation of the camera lens, but also registering images acquired by multiple uncalibrated similar sensor arrays as well as dissimilar sensors. In addition, subpixel accuracy may be necessary when the sensor arrays have different resolutions [1,3,14]. The intent of this research is to investigate the performance of a registration algorithm based on the projective transformation [5,13], which takes into account camera translation, rotation, pan, and tilt. This registration model is accepted as the most accurate of the available camera models, especially for reconnaissance and satellite imagery. Since it is a nonlinear model, direct least squares estimation is not capable of extracting the parameters properly from the data sets. In general, common feature points which appear within both the reference image and the image to be registered must be selected manually by the imaging specialist, and these features are used to aid in the parameter estimation algorithm. In this research, the block matching algorithm is used to automatically select correlated feature point pairs between two images, and these features are utilized in an iterative algorithm to estimate the nonlinear projective transformation parameters. Since the block matching algorithm is only capable of estimating accurate translation motion vectors in discontinuous edge regions, the inaccurate vectors are discarded by comparing the block matching feature points with the registered data computed using the currently estimated projective transformation parameters.

Experimental results show that the registration algorithm performs well on intensity image sequence frames, photographic and magnetic resonance (MR) slices from the Visible Human Project data set, and images acquired by a dual long-wave and short-wave infrared camera. Through subpixel-resolution registration, high-resolution video stills can be generated by integrating the registered pixels from a short sequence of low-resolution video frames. Simulations to enhance a short sequence of under-sampled intensity video frames show visual and quantitative improvement over the low-resolution reference frame. Multispectral segmentation techniques require that the images captured by dissimilar sensors are accurately registered. By exploiting the spectral correlations inherent between image channels and aligning the data properly, it is possible to extract spatial objects by partitioning clusters within the multidimensional feature space. Preliminary experiments are shown in utilizing dissimilar data registration followed by vector quantization [13] to segment tissues from multimodality Visible Human Project slices, as well as to enhance concealed weapons imaged by a dual long-wave and short-wave infrared camera.

This final report is organized as follows. In the Methodology section, the projective transformation model is introduced, and an iterative parameter estimation algorithm is presented which relies upon block matching to automatically select accurate feature point pairs between images. The Results section describes three simulations which verify the accuracy of integer- and subpixel-resolution registration using the projective transformation estimation algorithm, along with applications in multiframe integration and multispectral segmentation. A brief summary along with future directions is provided in the Conclusions section.

Methodology

The eight-parameter projective model [5,13] has been selected for use in the registration algorithm, since it can accurately account for camera translation, rotation, pan, and tilt. Before this nonlinear transformation is introduced, simpler linear motion models will be presented which are integral components of the projective model.

Assume that registration will be performed on two images, namely a reference image, denoted as $\mathbf{y}^{(k)}$, and a second image, denoted as $\mathbf{y}^{(l)}$, which contains similar information. The spatial information contained in the two images may be captured by the same camera after a transformation of the sensor array, two calibrated or uncalibrated cameras viewing the same scene, or two dissimilar sensor arrays. For a particular transformation model, the parameters must be estimated such that the pixels of $\mathbf{y}^{(l)}$ are warped to align with the pixels of $\mathbf{y}^{(k)}$ as well as possible. Perfect registration is rarely possible, due to either the inadequacy of the selected transformation model or independent object movement during the elapsed time between the capture of the two images.

Denote the point

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{1}$$

as a spatial location within image $\mathbf{y}^{(l)}$, and the corresponding point within the reference image $\mathbf{y}^{(k)}$ as

$$\mathbf{x}' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix}. \tag{2}$$

These two spatial locations correspond to a pixel value which is correlated between the images; *i.e.*, the pixel at location $\mathbf{x}$ in $\mathbf{y}^{(l)}$ undergoes a transformation to position $\mathbf{x}'$ in $\mathbf{y}^{(k)}$. By estimating the parameters of this transformation and then warping image $\mathbf{y}^{(l)}$ accordingly, the pixels should be registered between the two images. Obviously, a mathematical model must be selected for this transformation.

The simplest linear transformation is a pure displacement which occurs between the two images, denoted as

$$\mathbf{x}' = \mathbf{x} + \mathbf{b}. \tag{3}$$

This is the two-parameter displacement model, in which the translation vector,

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{4}$$

represents a shift of the camera sensor array. Although generally quite inaccurate, this model is by far the most popular, as it is used in video compression standards such as MPEG [6] and H.263 [9] to code the independent

motion of spatial pixel blocks. To estimate the parameters of the displacement model, the block matching algorithm [2,9] is typically used. In essence, a square block of pixels is selected within $\mathbf{y}^{(l)}$, and a search is performed to find the corresponding pixel block within the reference image. Provided that enough discontinuities are present within the selected block, this correlation technique results in an accurate displacement estimate. However, block matching fails in smooth spatial regions and areas of object occlusion. Popular matching criteria include the mean absolute error (MAE), the mean squared error (MSE), and the cross-correlation [9]. In this research, block matching is used to automatically select feature points and their transformations between the two images; these feature point pairs are then further processed to estimate the projective transformation parameters. Only the MSE and cross-correlation criteria will be utilized for this purpose. The block matching algorithm which uses the MSE criterion [9] is given as

$$\hat{\mathbf{b}}(\mathbf{x}) = \arg\min_{b_1,b_2} \frac{1}{(2p+1)^2} \sum_{m=x_1-p}^{x_1+p} \sum_{n=x_2-p}^{x_2+p} \left(y^{(l)}_{m,n} - y^{(k)}_{m+b_1,n+b_2} \right)^2, \tag{5}$$

and it is applicable when the two images have been acquired by similar, calibrated sensor arrays that result in constant pixel values for the same object. In the more general case of dissimilar sensor arrays or uncalibrated cameras which result in different pixel values for the same object, the block matching algorithm based on the cross-correlation criterion [9],

$$\hat{\mathbf{b}}(\mathbf{x}) = \arg\max_{b_1,b_2} \frac{\left| \sum_{m=x_1-p}^{x_1+p} \sum_{n=x_2-p}^{x_2+p} \left(y^{(l)}_{m,n} - \mu^{(l)}_{\mathbf{x}} \right) \left(y^{(k)}_{m+b_1,n+b_2} - \mu^{(k)}_{\mathbf{x}+\mathbf{b}} \right) \right|}{\sqrt{\sum_{m=x_1-p}^{x_1+p} \sum_{n=x_2-p}^{x_2+p} \left(y^{(l)}_{m,n} - \mu^{(l)}_{\mathbf{x}} \right)^2 \sum_{m=x_1-p}^{x_1+p} \sum_{n=x_2-p}^{x_2+p} \left(y^{(k)}_{m+b_1,n+b_2} - \mu^{(k)}_{\mathbf{x}+\mathbf{b}} \right)^2}}, \tag{6}$$

may be useful. In this expression, the absolute value of the correlation coefficient is used for robustness with respect to pixel value differences, with the local means defined as follows

$$\mu^{(l)}_{\mathbf{x}} = \frac{1}{(2p+1)^2} \sum_{m=x_1-p}^{x_1+p} \sum_{n=x_2-p}^{x_2+p} y^{(l)}_{m,n} \qquad \mu^{(k)}_{\mathbf{x}+\mathbf{b}} = \frac{1}{(2p+1)^2} \sum_{m=x_1-p}^{x_1+p} \sum_{n=x_2-p}^{x_2+p} y^{(k)}_{m+b_1,n+b_2} \tag{7}$$

For both the MSE and cross-correlation versions of the block matching algorithm, the square pixel block selected in $\mathbf{y}^{(l)}$ and centered at location $\mathbf{x}$,

$$\left\{ y^{(l)}_{m,n} \middle| \; x_1 - p \le m \le x_1 + p; x_2 - p \le n \le x_2 + p \right\}, \tag{8}$$

consists of $(2p + 1)$ x $(2p + 1)$ image pixels, and the square search area in the reference image,

$$\left\{ y^{(k)}_{m,n} \middle| \; x_1 - d \le m \le x_1 + d; x_2 - d \le n \le x_2 + d \right\}, \tag{9}$$

allows for a maximum displacement of d pixels in either direction. As rules of thumb, larger pixel blocks may be more accurately correlated between frames, and larger search areas allow for long-range motion, but the block matching algorithm becomes highly computational for large values of p and d. In practice, a value of p=4 results in relatively accurate displacement estimates. In the presence of a global displacement occurring between the images undergoing registration, block matching can be applied at sample points throughout the data, with the results either averaged or α-trimmed mean filtered [11] to provide the single translation vector.

The six-parameter linear affine model,

$$\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{b}\,, \tag{10}$$

takes into account both image displacements and rotations. The 2 x 2 rotation matrix,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \tag{11}$$

is combined with the two-parameter translation vector from the displacement model to provide a more realistic transformation. The parameters of this linear model can be estimated using least squares,

$$\hat{\mathbf{A}}, \hat{\mathbf{b}} = \arg\min_{\mathbf{A},\mathbf{b}} \sum_{\mathbf{x}} \left\| \mathbf{x}' - \mathbf{A}\mathbf{x} - \mathbf{b} \right\|^2, \tag{12}$$

provided that at least several feature points $\mathbf{x}$ in $\mathbf{y}^{(l)}$ and the corresponding transformed feature locations $\mathbf{x}'$ in $\mathbf{y}^{(k)}$ are selected by the user.

Estimating the parameters of the two linear models can be performed rather efficiently and accurately, but they do not account for the additional real-world transformations of camera pan and tilt. To accommodate for these perspective transformations, the eight-parameter projective model [5,13],

$$\mathbf{x}' = \frac{\mathbf{A}\mathbf{x} + \mathbf{b}}{\mathbf{c}^t \mathbf{x} + 1}, \tag{13}$$

has been selected for the registration algorithm. The numerator of the projective model is the linear six-parameter affine model, which accounts for translation and rotation, while camera pan and tilt is modeled by the vector

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{14}$$

in the denominator. The notation $\mathbf{c}^t$ denotes a vector transpose. Unfortunately, the nonlinear nature of this model makes estimation of the eight parameters difficult, and an iterative technique must be developed.

To estimate the projective transformation parameters, an iterative least squares [8] approach will be taken. The least squares problem statement is defined as

$$\hat{\mathbf{A}}, \hat{\mathbf{b}}, \hat{\mathbf{c}} = \arg\min_{\mathbf{A},\mathbf{b},\mathbf{c}} \sum_{\mathbf{x}} \left\| \mathbf{x}' \mathbf{c}^t \mathbf{x} + \mathbf{x}' - \mathbf{A}\mathbf{x} - \mathbf{b} \right\|^2. \tag{15}$$

The block matching algorithm will be used to automatically select feature points $\mathbf{x}$ in $\mathbf{y}^{(l)}$ and their transformations $\mathbf{x}'$ in $\mathbf{y}^{(k)}$ during each iteration. For the i^{th} iteration of the algorithm, a least squares estimate of the transformation parameters, $(\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i)$, is computed from a set of N feature points and their transformations, $(\mathbf{x}_k, \mathbf{x}'_k)$ for k=1,...,N, as image $\mathbf{y}^{(l)}$ is iteratively warped towards $\mathbf{y}^{(k)}$. A recursive relationship for the overall transformation can be expressed as

$$\mathbf{x}' = \frac{\left[\mathbf{A}_i \mathbf{A}_{i-1} + \mathbf{b}_i \mathbf{c}_{i-1}^t\right]\mathbf{x} + \left[\mathbf{b}_i + \mathbf{A}_i \mathbf{b}_{i-1}\right]}{\left[\mathbf{c}_i^t \mathbf{A}_{i-1} + \mathbf{c}_{i-1}^t\right]\mathbf{x} + 1}. \tag{16}$$

If the images to be registered have similar appearances, convergence is generally achieved in two to three iterations.

An important question arises: how does the user select the feature points in the parameter estimation algorithm? Furthermore, how many points are required, and is it possible to perform the feature point selection process automatically? The user looks for N feature points $\mathbf{x}_k$ in $\mathbf{y}^{(l)}$ which are spaced far apart and do not lie on the same line. These points should correspond to pertinent objects within the scene. For every feature point $\mathbf{x}_k$ selected in $\mathbf{y}^{(l)}$, the corresponding transformed point $\mathbf{x}'_k$ must be located within the reference image $\mathbf{y}^{(k)}$. These feature point pairs, $(\mathbf{x}_k, \mathbf{x}'_k)$ for k=1,...,N, are then used to estimate the transformation parameters. Typically, at least N=4 feature points pairs must be selected manually by the user. By selecting additional features, the user must spend more time working directly on the images to be registered, but the projective transformation parameter estimates should improve. It is possible to select the feature points automatically by sparsely sampling the data in a regular pattern. For example, every fourth point could be selected both horizontally and vertically, so that 1/16-th of the pixels within $\mathbf{y}^{(l)}$ are used as features. Automatically computing the transformation of each point is rather challenging, as only an initial guess can be made at the positions of $\mathbf{x}'_k$ for k=1,...N within the reference image $\mathbf{y}^{(k)}$. The block matching algorithm can be used to generate the initial estimates of these transformed positions, with the knowledge that many of the feature points located in smooth image regions will yield inaccurate feature point pairs. By first detecting and eliminating these M poor motion vectors and then calculating a least squares solution for the projective transformation parameters using the (N-M) remaining feature point pairs, it is possible to estimate the transformation in most cases both automatically and efficiently.

The eight-parameter projective transformation algorithm is an iterative estimation technique, in which $\mathbf{y}^{(l)}$ is recursively warped towards the reference image $\mathbf{y}^{(k)}$ as the transformation parameters converge to their true values. At every step of the algorithm, block matching is used to automatically select the feature point pairs, since the transformed feature points will change their positions as $\mathbf{y}^{(l)}$ is progressively warped. The projective transformation algorithm is described as follows:

Eight-Parameter Projective Transformation Estimation Algorithm

1. Set $\mathbf{y}_0^{(l)} = \mathbf{y}^{(l)}$, $\mathbf{A}_0$=$\mathbf{I}$, $\mathbf{b}_0$=$\mathbf{0}$, and $\mathbf{c}_0$=$\mathbf{0}$. Set the iteration number i=1.
2. Select N feature points $\mathbf{x}_k$, for k=1,...,N, by sparsely sampling a region within image $\mathbf{y}_i^{(l)}$ which contains "spatially active" image information; *i.e.*, areas containing large numbers of edges. Every fourth point may be selected both horizontally and vertically within a spatially active region to achieve acceptable results.
3. Estimate block matching motion vectors at each of the N selected feature points, using the MSE criterion for images captured by similar sensor arrays or the cross-correlation criterion for uncalibrated and dissimilar sensor images. Let the k^{th} transformed point be denoted as

$$\mathbf{x}'_k = \mathbf{x}_k + \hat{\mathbf{b}}(\mathbf{x}_k) .$$

4. Estimate the projective transformation parameters, $(\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i)$, using all N block matching feature point pairs $(\mathbf{x}_k, \mathbf{x}'_k)$ to calculate the least squares solution [8] to the following problem statement:

$$\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i = \arg\min_{\mathbf{A},\mathbf{b},\mathbf{c}} \sum_{k=1}^{N} \left\| \mathbf{x}'_k \mathbf{c}^t \mathbf{x}_k + \mathbf{x}'_k - \mathbf{A}\mathbf{x}_k - \mathbf{b} \right\|^2$$

5. Since the block matching vectors located in smooth image regions and areas of object occlusion will be inaccurate, these poor feature point pairs will be statistically eliminated from the projective transformation least squares solution. Denote the residual of the k^{th} transformed feature point, $\mathbf{x}'_k$, and the currently estimated affine portion of the projective model, $\mathbf{A}_i \mathbf{x}_k + \mathbf{b}_i$, as

$$\mathbf{r}_k = \begin{bmatrix} r_{k_1} \\ r_{k_2} \end{bmatrix} = \begin{bmatrix} x'_{k_1} - a_{i_{11}} x_{k_1} - a_{i_{12}} x_{k_2} - b_{i_1} \\ x'_{k_2} - a_{i_{21}} x_{k_1} - a_{i_{22}} x_{k_2} - b_{i_2} \end{bmatrix}.$$

Calculate the sample mean,

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{r}_k,$$

and sample variances of the residuals, given by the expressions

$$\sigma_1^2 = \frac{1}{N-1} \sum_{k=1}^{N} \left(x'_{k_1} - a_{i_{11}} x_{k_1} - a_{i_{12}} x_{k_2} - b_{i_1} - \mu_1 \right)^2 \qquad \sigma_2^2 = \frac{1}{N-1} \sum_{k=1}^{N} \left(x'_{k_2} - a_{i_{21}} x_{k_1} - a_{i_{22}} x_{k_2} - b_{i_2} - \mu_2 \right)^2.$$

If $|r_{k_1}| > \sigma_1$ or $|r_{k_2}| > \sigma_2$, eliminate the corresponding feature point pair $(\mathbf{x}_k, \mathbf{x}'_k)$ from the least squares estimation problem. By conducting this test for all feature point pairs $(\mathbf{x}_k, \mathbf{x}'_k)$, k=1,…,N, the M least accurate block matching features will be detected and eliminated.

6. Re-estimate the projective transformation parameters, $(\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i)$, using the $(N - M)$ most accurate block matching feature point pairs to calculate the least squares solution to the following problem statement:

$$\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i = \arg\min_{\mathbf{A},\mathbf{b},\mathbf{c}} \sum_{k=1}^{N-M} \left\| \mathbf{x}'_k \mathbf{c}^t \mathbf{x}_k + \mathbf{x}'_k - \mathbf{A}\mathbf{x}_k - \mathbf{b} \right\|^2$$

7. Warp the original image $\mathbf{y}^{(l)}$ by applying the overall projective transformation,

$$\mathbf{x}' = \frac{\left[\mathbf{A}_i \mathbf{A}_{i-1} + \mathbf{b}_i \mathbf{c}_{i-1}^t\right]\mathbf{x} + \left[\mathbf{b}_i + \mathbf{A}_i \mathbf{b}_{i-1}\right]}{\left[\mathbf{c}_i^t \mathbf{A}_{i-1} + \mathbf{c}_{i-1}^t\right]\mathbf{x} + 1},$$

to all pixels, and set $\mathbf{y}_i^{(l)}$ equal to the resulting image. This warped image is to be registered with the reference image $\mathbf{y}^{(k)}$ during the next iteration.

8. Calculate the change in the projective model coefficients from iteration i-1 to i. If the change is small, the algorithm has converged, and the aggregate parameter estimates are given as follows:

$$\hat{\mathbf{A}} = \mathbf{A}_i \mathbf{A}_{i-1} + \mathbf{b}_i \mathbf{c}_{i-1}^t \qquad \hat{\mathbf{b}} = \mathbf{b}_i + \mathbf{A}_i \mathbf{b}_{i-1} \qquad \hat{\mathbf{c}} = \mathbf{A}_{i-1}^t \mathbf{c}_i + \mathbf{c}_{i-1}.$$

Otherwise, set i=i+1, and return to Step 2. Convergence is generally achieved in two to three iterations.

The projective transformation estimation algorithm is directly applicable to integer-resolution image registration. In the case of subpixel-resolution registration, the reference image, $\mathbf{y}^{(k)}$, and the image to be registered, $\mathbf{y}^{(l)}$, must both be up-sampled by a factor of q. The up-sampled images are then used directly in the projective transformation algorithm to estimate the $1/q$-th pixel resolution parameters. The primary methods of image interpolation which are available, in order of both increasing accuracy and computational complexity, include zero-order hold up-sampling, bilinear interpolation, cubic B-spline interpolation [4], and variational methods such as Bayesian maximum *a posteriori* (MAP) estimation [10]. In general, more accurate up-sampled images result in more accurate subpixel-resolution transformation parameter estimates. In the simulations conducted for this report, up-sampling is performed using cubic B-spline interpolation, in which localized third-order polynomials known as B-splines are fit to the image data [4]. This method provides relatively accurate interpolated images at a reasonable computational cost.

Results

Several experiments were conducted to show the efficacy of the integer-resolution and subpixel-resolution registration algorithms. Different types of data sets were selected to show not only the robustness of the technique, but also how the parameter estimation method may fail on extremely low-resolution images and images acquired by dissimilar sensors which do not possess correlated discontinuities. The three different data sets include the following:

- *Film* Image Sequence – Frames 15 and 18 from an intensity image sequence were selected as an example of a data set captured by a single sensor array after a transformation of the camera lens. The entire image sequence is a film taken by a news crew from a helicopter, and it contains translational and rotational motion as well as a slight camera pan and tilt. Since the image sequence was captured by a single camera, the object intensities between frames should be relatively constant.
- *Visible Human Project* Data Set – The Visible Human Project is an extensive set of medical images managed by the National Library of Medicine. Within this data set, three biomedical imaging modalities were used to image an entire male cadaver and an entire female cadaver. Computed tomography (CT) image slices were acquired to provide high-definition resolution of the skeletal system, and magnetic resonance (MR) image slices were acquired which accurately show the soft tissues of the bodily organs. The MR data consists of three image channels per slice, including the PD, T1, and T2 images. The cadavers were frozen cryogenically, and color (RGB) photographs were taken of the bodies as the tissue was microtomed from head to toe. The luminance of one cryogenic photographic image slice and the corresponding PD channel of an MR slice in the head region of the Visible Male were used in these experiments, representing two images acquired by dissimilar sensors.
- *Concealed Weapon* Images – A dual long-wave and short-wave infrared camera was used to capture images of a person carrying a concealed handgun, to determine whether an automated algorithm could be developed to

detect and enhance the weapon. This data set was used as an example of images captured by uncalibrated cameras.

Figures 1 and 2 show the results of the integer-resolution and subpixel-resolution registration algorithm on the *Film* image sequence, respectively. In the subpixel case, the original frames were first down-sampled by a factor of q=4, and then up-sampled to their original sizes using cubic B-spline interpolation. Both the MSE and cross-correlation measures were used to estimate the block matching feature point pairs, to show how well each matching criterion performs on the intensity data. To show the quality of the registration, the reference frame and the second warped frame have been superimposed, since there is no meaningful quantitative registration quality measure. In both the integer-resolution and subpixel-resolution registrations, the data seems to align extremely well, with the exception of the spire near the middle of the frames and the vehicles which have moved during the sampling interval. Since the video frames were acquired by the same camera, the intensity values are relatively constant for objects throughout the image sequence; therefore, the MSE and cross-correlation criteria perform equally well.

As an example of an application in which subpixel registration may be beneficial, multiframe enhancement [7,12,15] has been performed on the down-sampled *Film* image sequence. Multiframe enhancement involves the integration of a short sequence of $(2N + 1)$ low-resolution video frames, denoted as $\{\mathbf{y}^{(k-N)}, \mathbf{y}^{(k-N+1)}, \ldots, \mathbf{y}^{(k-1)}, \mathbf{y}^{(k)}, \mathbf{y}^{(k+1)}, \ldots, \mathbf{y}^{(k+N-1)}, \mathbf{y}^{(k+N)}\}$, to generate an estimate of the original high-resolution video still (HRVS) image $\mathbf{z}^{(k)}$ located at the center of the sequence. By registering a set of low-resolution image sequence frames with respect to a reference frame, the pixels should be aligned. However, the low-resolution pixels may not be perfectly registered on a subpixel-resolution grid, and this subpixel overlap can be exploited to estimate the values of the high-resolution still image and reduce aliasing artifacts. To perform multiframe enhancement, all candidate frames are first up-sampled by a factor of q using cubic B-spline interpolation. Using the cubic B-spline interpolated data, subpixel-resolution projective transformation parameters are estimated for every frame with respect to the reference image. Each original low-resolution frame is then expanded by a factor of q using zero-order hold up-sampling, and these blocky images are warped using the estimated subpixel-resolution projective transformation parameters. A linear or nonlinear operation may be performed on the pixels throughout the ensemble of registered frames to estimate the high-resolution pixel value at a particular point. Averaging the $(2N + 1)$ pixels at position $\mathbf{x}$ within the zero-order hold up-sampled, registered sequence $\{\mathbf{y}^{(k-N)}, \mathbf{y}^{(k-N+1)}, \ldots, \mathbf{y}^{(k-1)}, \mathbf{y}^{(k)}, \mathbf{y}^{(k+1)}, \ldots, \mathbf{y}^{(k+N-1)}, \mathbf{y}^{(k+N)}\}$ tends to result in a blurring of the data. Rather, a vector median filter is applied to these aligned pixels, with the filter output used as the value of the high-resolution image estimate $\hat{\mathbf{z}}^{(k)}$ at location $\mathbf{x}$. Figure 3 depicts the results of the multiframe integration simulations on 3, 5, and 7 down-sampled frames from the *Film* image sequence. Visually, the HRVS estimate computed using 5 frames appears to be the closest to the original high-resolution reference image. Quantitatively, the peak signal-to-noise ratio for grayscale (8-bit) image data can be used to measure the distance between the video still estimates and the original high-resolution test reference image. This value, in decibels, is given as

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{N_1 N_2} \left\| \mathbf{z}^{(k)} - \hat{\mathbf{z}}^{(k)} \right\|^2},$$

where 255 represents the maximum value of an 8-bit image pixel, and the images contain N_1 x N_2 pixels. The multiframe estimates all have higher PSNR values than the down-sampled reference frame expanded using the zero-order hold. An advantage of this multiframe enhancement technique over other super-resolution enhancement algorithms [7,12,15] is that the original image sequence frames are not interpolated prior to their integration. Zero-order hold up-sampling simply replicates the pixels within a square block, and these large pixels are then warped using the estimated transformation parameters. No new image information is incorporated into the high-resolution video still estimate; the data is used directly without alteration.

Integer- and subpixel-resolution Visible Human Project image slices have been registered, with the results shown in Figures 4 and 5. Block matching feature point pairs were only computed in the head region of the image slices, since the black background would result in poor estimates. Since the two images were acquired by different scanners, the pixel values are very different and the edges are not necessarily correlated between the slices. As expected, the MSE block matching criterion is not capable of estimating accurate feature point pairs for these dissimilar sensor images. The advantage of the cross-correlation criterion is apparent in this experiment, as the correlation coefficient is robust with respect to pixel value differences. It should be noted that the cryogenic photographic image was used as the reference image in the registration algorithm; this image was selected due to its rich edge information as compared to the PD channel of the MR data. Simulations were attempted in which each image slice was used as the reference. A successful integer-resolution registration resulted for either reference image using the cross-correlation criterion. In the subpixel-resolution experiments, however, a successful registration resulted for the cryogenic photographic reference image but not for the PD channel reference. The lack of structure in the down-sampled PD channel does not provide enough information for use as the reference image.

Multispectral segmentation is a technique in which the spectral correlations are exploited in a registered data set to segment objects within a scene. As shown in Figure 4, the registered Visible Human Project data was segmented into K=9 different classes using vector quantization. The K-means algorithm [9] was used for this purpose, in which a uniform quantization of the two-dimensional feature space (histogram) provided the initial condition, and the iterative algorithm converged to a suboptimal partition of the feature space clusters. As expected, vector quantization provides accurate results for properly registered data, but it fails on the misregistered image slices.

In the third set of experiments, the *Concealed Weapon* infrared images were registered on the integer-resolution grid to possibly improve the alignment of the handgun. As shown in Figure 6, large regions of the data set are smooth. For this reason, block matching feature point selection was restricted to a small window surrounding the handgun. In Figure 7, it is not conclusive which block matching criteria, MSE or cross-correlation, performs better on these images. Since the cross-correlation criterion is more robust with respect to pixel differences, it is generally the more trust-worthy criterion. Again, the K-means algorithm was applied to the registered data sets with the intent of segmenting the handgun from the scene. Although not evident in the luminance versions of the segmented images, the handgun is classified with a unique color. The registration of multiple infrared camera images followed by vector quantization represents a promising method of semi-automated concealed weapon enhancement.

Conclusion

An image registration algorithm based on the eight-parameter projective model has been investigated which accounts for camera translation, rotation, pan, and tilt. Automated feature selection is carried out through the block matching algorithm, in which the end-points of translation vectors serve as feature point pairs. An iterative least squares solution to the projective transformation parameters is calculated from these features, and the image to be registered is progressively warped towards the reference image. Simulations on intensity image sequence frames, biomedical imagery acquired by dissimilar sensors, and uncalibrated infrared camera images show that the technique results in accurate integer- and subpixel-resolution registrations, and that the algorithm is fairly robust with respect to differences between data types.

As far as future work is concerned, the algorithm should be tested on a wider variety of data sets to determine its utility. Currently, the user must specify a window for the feature points, so that block matching does not estimate erroneous vectors on irrelevant data such as the background of the biomedical image slices. To make the registration algorithm truly automated, a method of selecting the proper region could be developed. Block matching using the cross-correlation criterion seems to select accurate feature point pairs between dissimilar images, provided that the two images possess some common edges. If possible, it may be advantageous to examine matching algorithms which do not necessarily rely on correlated discontinuities between the images to be registered but rather on object contours. In addition, the robustness of the subpixel-resolution registrations must be examined with respect to data sparsity. Further research will also be conducted in the multiframe integration of registered image sequence frames; this technique may be useful in the realm of dissimilar sensor fusion as well.

Acknowledgments

The author wishes to thank the Air Force Office of Scientific Research Summer Faculty Research Program and Mr. Mark G. Alford of Rome Research Site for providing the opportunity and the facilities for conducting this research. The author also wishes to acknowledge the National Library of Medicine for providing the Visible Human Project license agreement, and Dr. Richard A. Robb, Director of the Biomedical Imaging Resource at the Mayo Foundation and Clinic, for providing the actual biomedical image data.

References

[1] C. A. Berenstein, L. N. Kanal, D. Lavine, and E. C. Olson, "A Geometric Approach to Subpixel Registration Accuracy," *Computer Vision, Graphics, and Image Processing*, volume 40, pages 334-360, 1987.

[2] B. Furht, J. Greenberg, and R. Westwater, *Motion Estimation Algorithms for Video Compression*. Kluwer Academic Publishers, 1997.

[3] G. de Haan and P. W. A. C. Beizen, "Sub-pixel Motion Estimation with 3-D Recursive Search Block-Matching," *Signal Processing: Image Communications*, volume 6, pages 229-239, 1994.

[4] H. H. Hou and H. C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 26, number 6, pages 508-517, 1978.

[5] S. Mann and R. W. Picard, "Virtual Bellows: Constructing High Quality Stills from Video," in *Proceedings of the IEEE International Conference on Image Processing*, (Austin, TX), pages 363-367, 1994.

[6] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York, NY: Chapman & Hall, 1996.

[7] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "High-Resolution Standards Conversion of Low-Resolution Video," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Detroit, MI), pages 2197-2200, 1995.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition. Cambridge University Press, 1992.

[9] K. R. Rao, and J. J. Hwang, *Techniques & Standards for Image, Video & Audio Coding*. Upper Saddle River, NJ: Prentice Hall PTR, 1996.

[10] R. R. Schultz and R. L. Stevenson, "A Bayesian Approach to Image Expansion for Improved Definition," *IEEE Transactions on Image Processing*, volume 3, number 3, pages 233-242, 1994.

[11] R. R. Schultz and R. L. Stevenson, "Estimation of Subpixel-Resolution Motion Fields from Segmented Image Sequences." Presented at SPIE's 12th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (AeroSense '98). Conference Title: Sensor Fusion: Architectures, Algorithms, and Applications II, (Orlando, FL), April 13-17, 1998.

[12] R. R. Schultz and R. L. Stevenson, "Extraction of High-Resolution Frames from Video Sequences," *IEEE Transactions on Image Processing*, volume 5, number 6, pages 996-1101, 1996.

[13] A. M. Tekalp, *Digital Video Processing*. Upper Saddle River, NJ: Prentice-Hall, Inc., 1995.

[14] Q. Tian and M. N. Huhns, "Algorithms for Subpixel Registration," *Computer Vision, Graphics, and Image Processing*, volume 35, pages 220-233, 1986.

[15] R. Y. Tsai and T. S. Huang, "Multiframe Image Restoration and Registration," in *Advances in Computer Vision and Image Processing*, (R. Y. Tsai and T. S. Huang, eds.), volume 1, pages 317-339, JAI Press Inc., 1984.

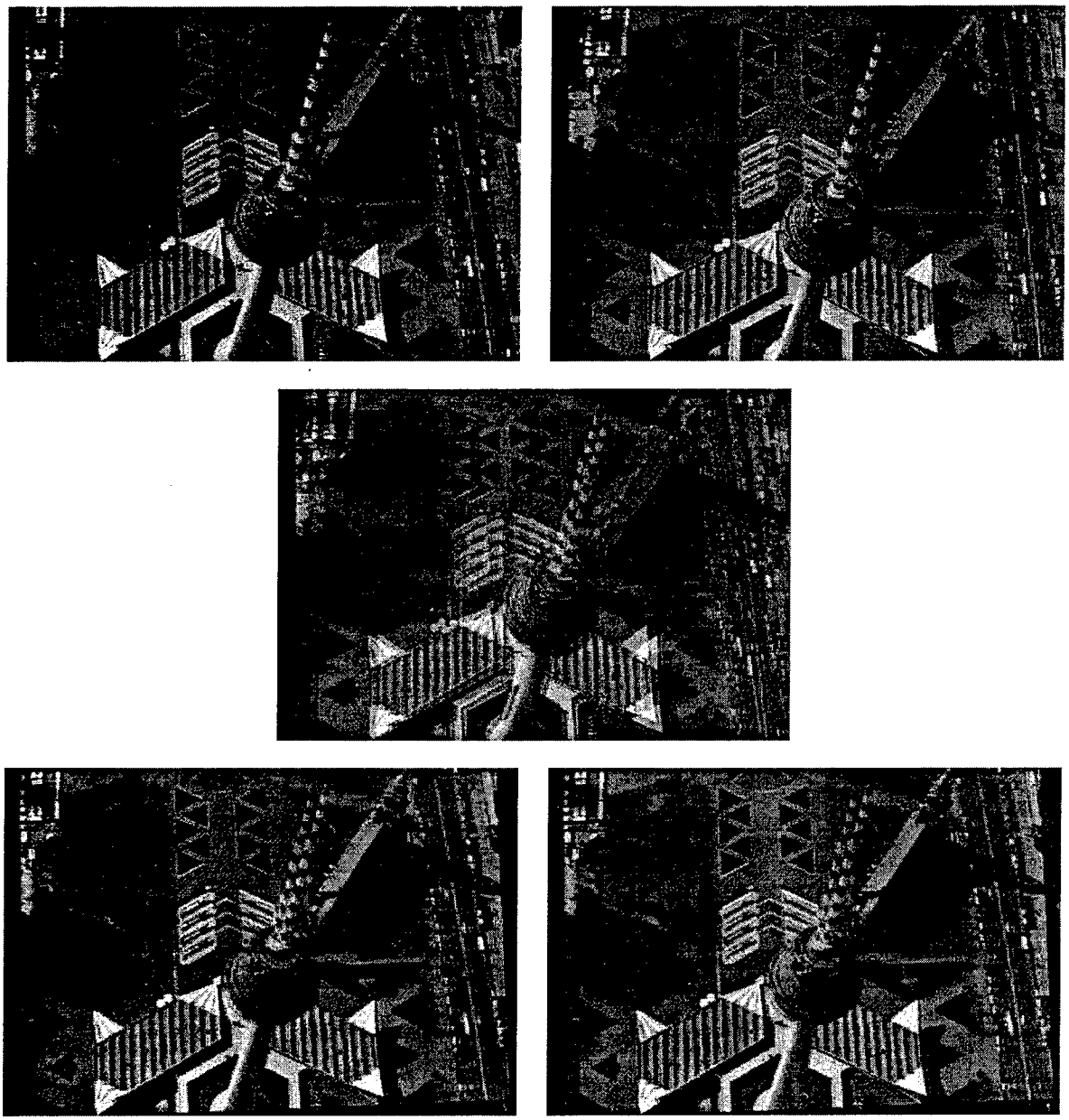


Figure 1: Integer-resolution registration of two frames from the *Film* image sequence. *Top Row, Left-to-Right: Film* sequence frames 15 (reference) and 18. *Second Row:* Superimposed unregistered frames. *Third Row, Left-to-Right:* Superimposed registered frames computed using block matching feature point pair selection with the MSE criterion, and the superimposed registered frames calculated using block matching feature point pair selection with the cross-correlation criterion. No visible difference results from the use of either block matching criterion.

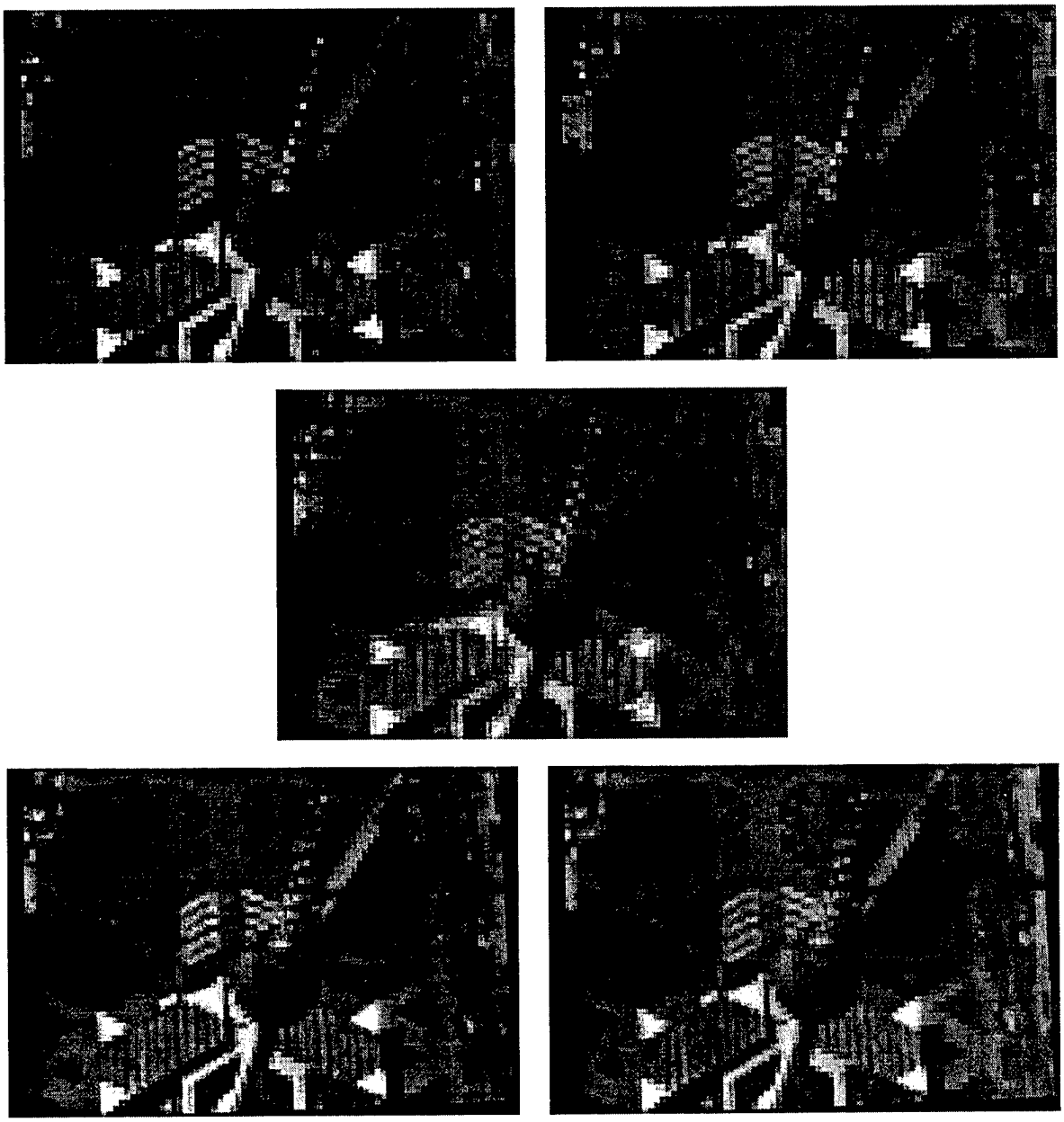


Figure 2: Subpixel-resolution registration of two frames from the *Film* image sequence. *Top Row, Left-to-Right: Film* sequence frames 15 (reference) and 18; both frames have been down-sampled by a factor of q=4. *Second Row:* Superimposed unregistered frames. *Third Row, Left-to-Right:* Superimposed registered frames computed using block matching feature point pair selection with the MSE criterion, and the superimposed registered frames calculated using block matching feature point pair selection with the cross-correlation criterion. No visible difference results from the use of either block matching criterion.

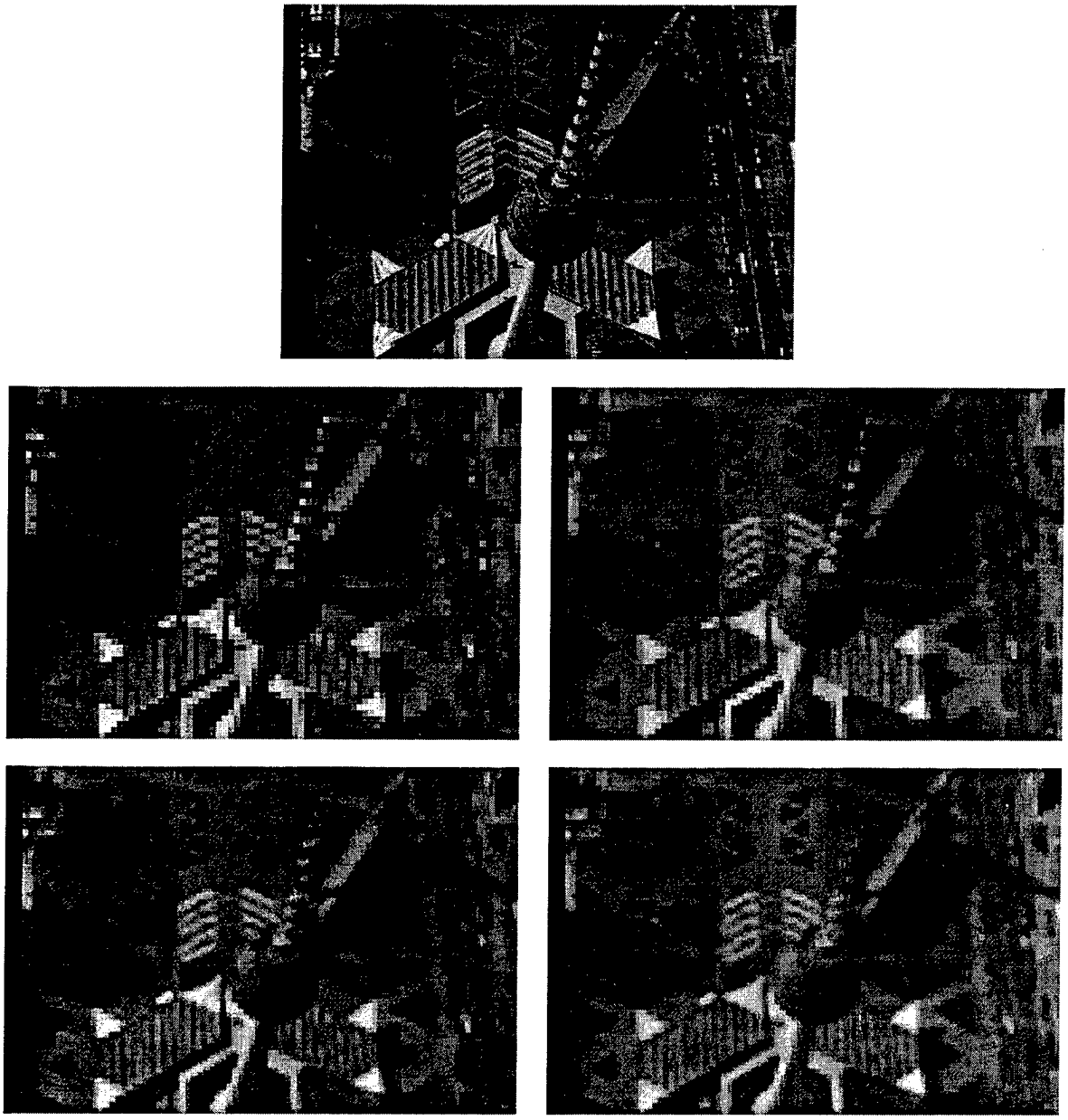


Figure 3: Multiframe integration of the *Film* image sequence. *First Row:* High-resolution *Film* frame 15, representing the exact video still image. *Second Row, Left-to-Right:* Reference frame 15, down-sampled by a factor of q=4 (PSNR=25.02 dB), and the high-resolution video still computed using three (3) adjacent down-sampled frames (PSNR=25.49 dB). *Third Row, Left-to-Right:* High-resolution video still calculated using five (5) adjacent down-sampled frames (PSNR=25.52 dB), and the video still estimated using seven (7) adjacent down-sampled frames (PSNR=25.43).

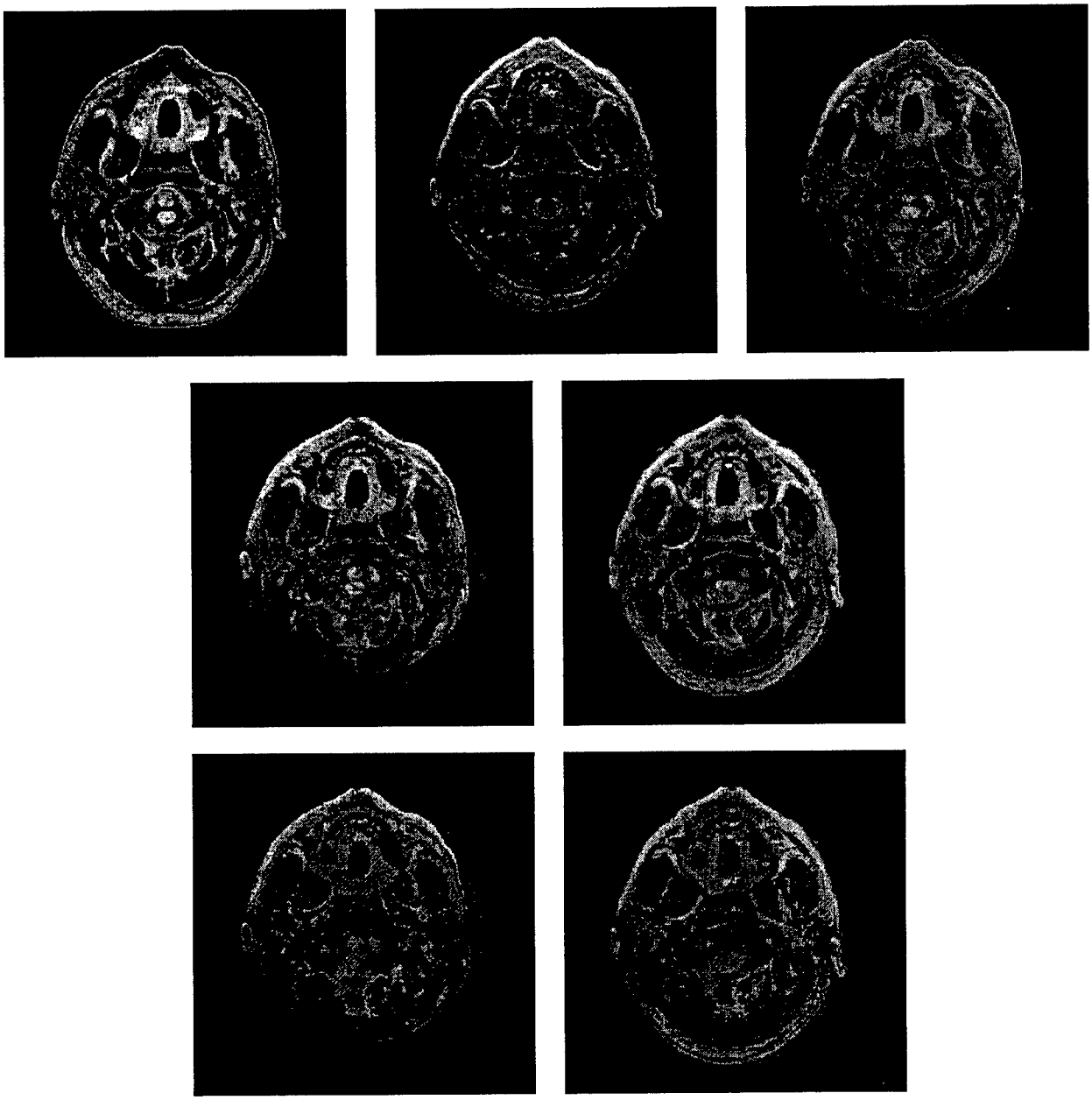


Figure 4: Integer-resolution registration of multimodality medical image slices from the *Visible Human Project* data set. *First Row, Left-to-Right:* Luminance of cryogenic photographic image slice (reference image), PD channel of the magnetic resonance (MR) slices, and the superimposed unregistered images. *Second Row, Left-to-Right:* Superimposed registered slices computed using block matching feature point pair selection with the MSE criterion, and the superimposed registered slices calculated using block matching feature point selection with the cross-correlation criterion. Note the greatly improved quality of the registration which results from using the cross-correlation criterion. *Third Row, Left-to-Right:* Output of a K-means vector quantizer (K=9) with the MSE registered images used as input, and the output of a K-means vector quantizer (K=9) with the cross-correlation registered image slices used as input.

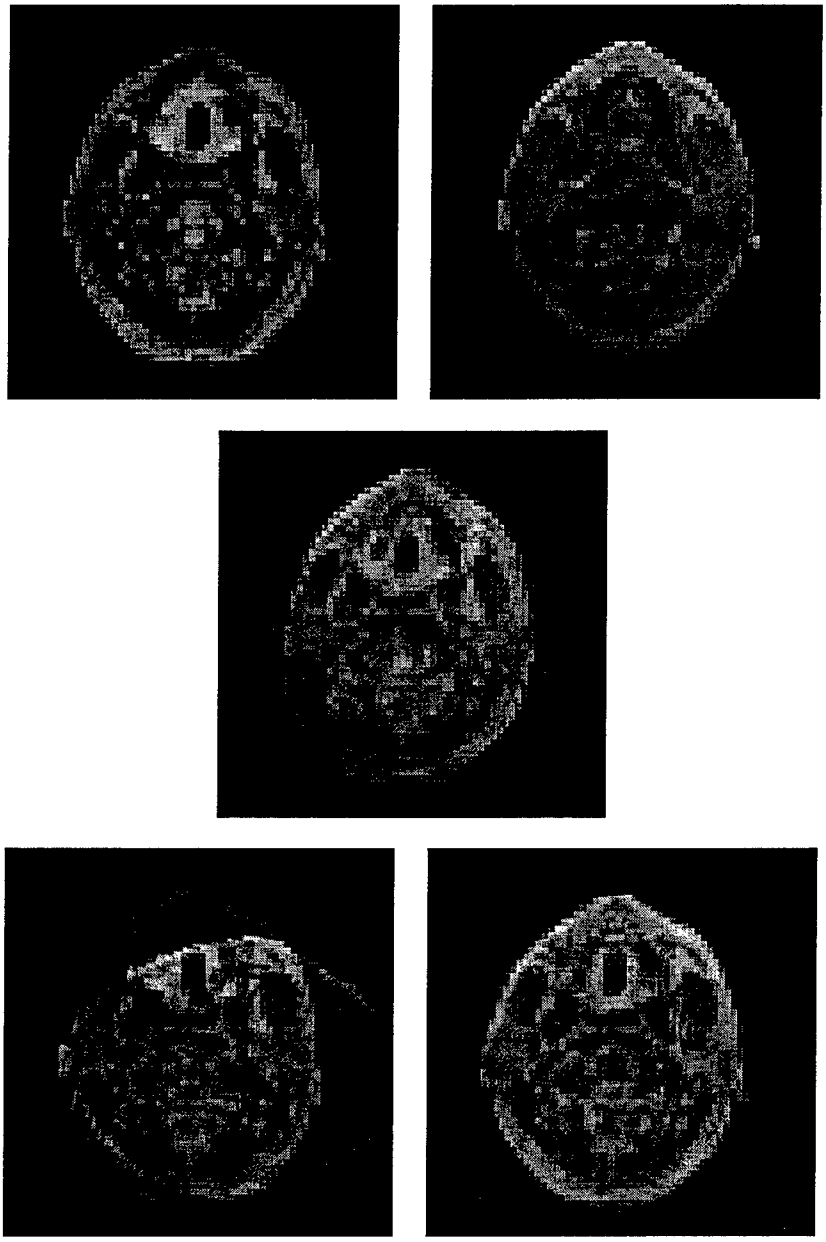


Figure 5: Subpixel-resolution registration of multimodality medical image slices from the *Visible Human Project* data set. *First Row, Left-to-Right:* Luminance of cryogenic photographic image slice (reference image), and PD channel of the magnetic resonance (MR) data. Both image slices were down-sampled by a factor of q=4. *Second Row:* Superimposed unregistered image slices. *Third Row, Left-to-Right:* Superimposed registered slices computed using block matching feature point pair selection with the MSE criterion, and the superimposed registered slices calculated using block matching feature point pair selection with the cross-correlation criterion. Note the greatly improved quality of the registration which results from using the cross-correlation criterion.

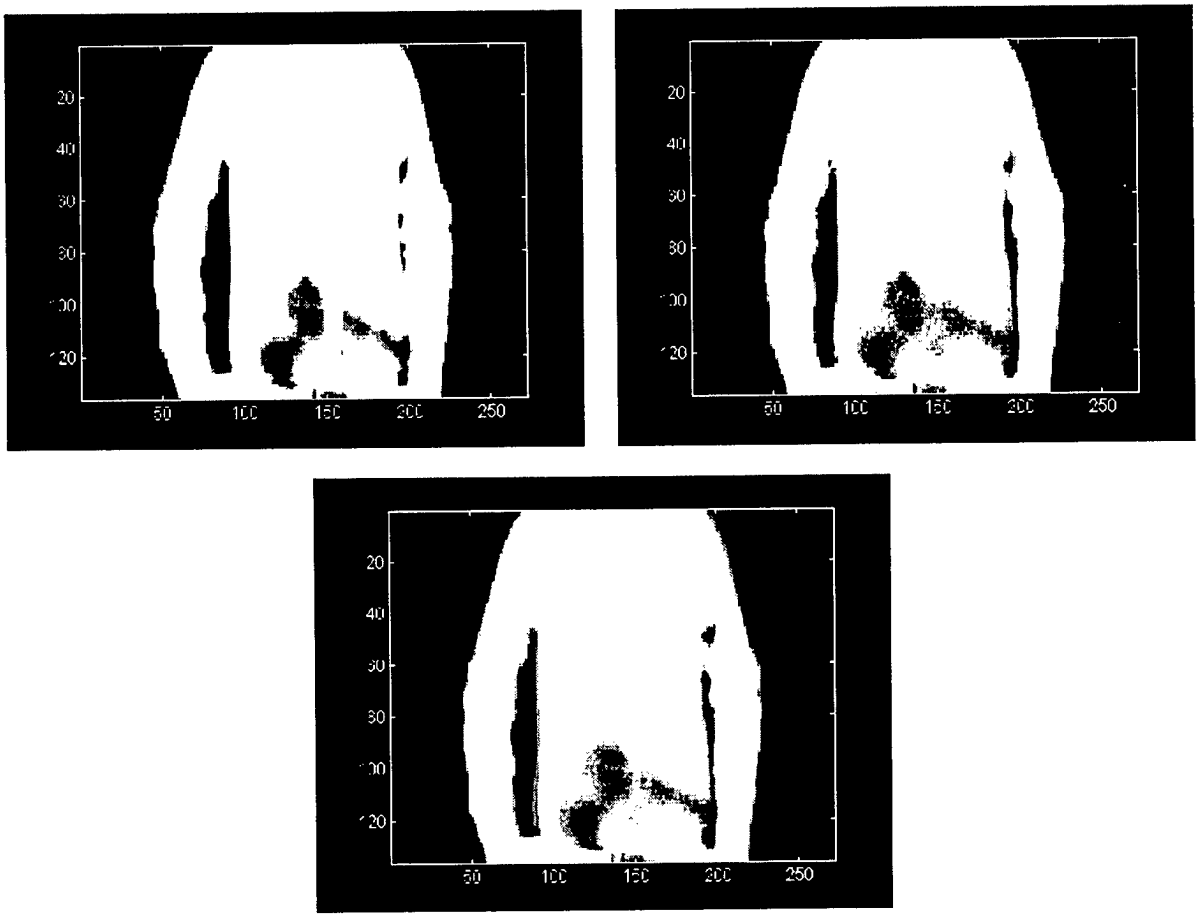


Figure 6: *Concealed Weapon* images acquired by a dual long-wave and short-wave infrared camera. *First Row, Left-to-Right:* Image captured by the long-wave infrared camera, and the image captured by the short-wave infrared camera. *Second Row:* Superimposed unregistered frames.

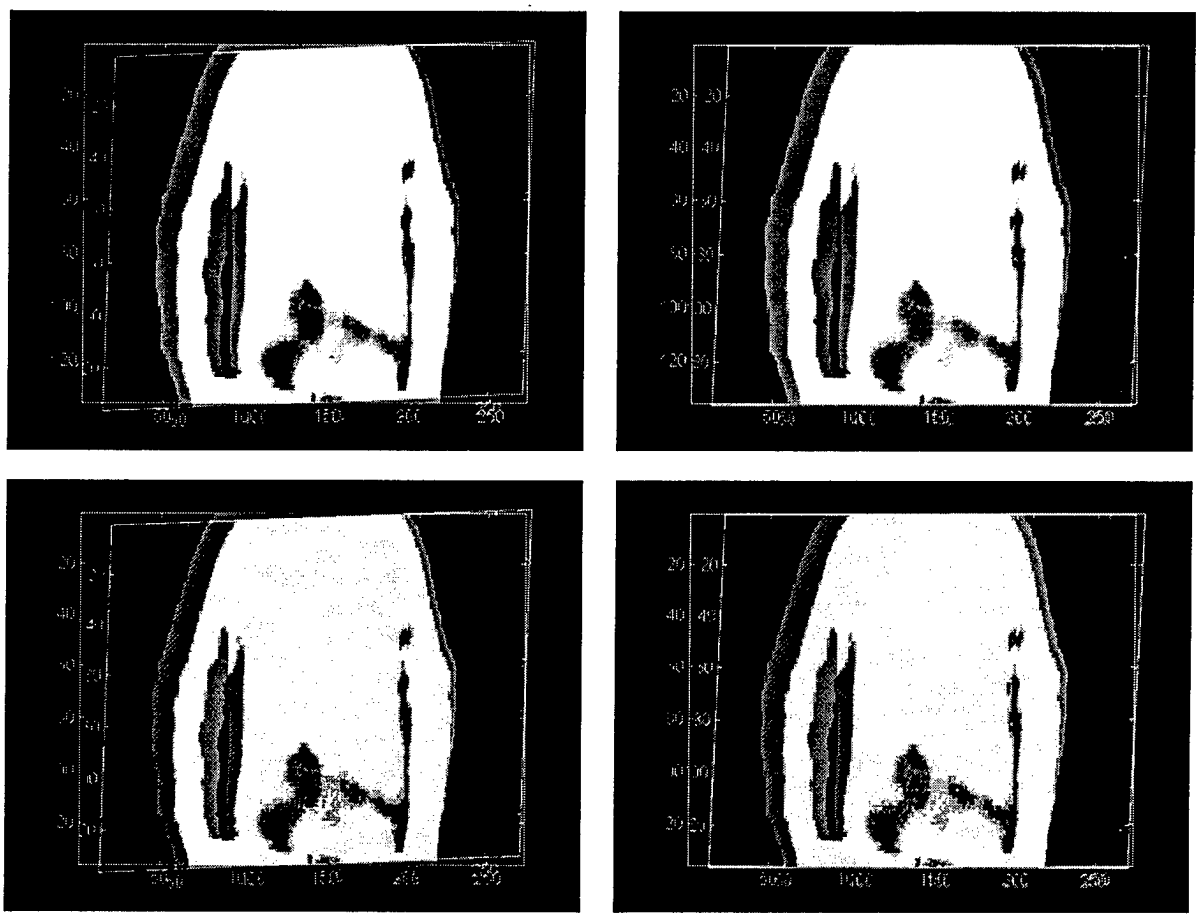


Figure 7: Registration and vector quantization of the *Concealed Weapon* infrared camera images. *First Row, Left-to-Right:* Superimposed registered slices computed using block matching feature point pair selection with the MSE criterion, and the superimposed registered slices calculated using block matching feature point pair selection with the cross-correlation criterion. Although the registration resulting from using the cross-correlation criterion seems to be more accurate, the lack of truth information does not allow for a definite analysis. *Second Row, Left-to-Right:* Output of a K-means vector quantizer (K=9) with the MSE registered images used as input, and the output of a K-means vector quantizer (K=9) with the cross-correlation registered image slices used as input.

ENHANCEMENTS TO CUBEWORLD

Kalpathi R. Subramanian
Assistant Professor
Department of Computer Science

The University of N. Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC 28223-0001

Final Report for:
Summer Faculty Research Program
Rome Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Rome Research Site

August 1998

ENHANCEMENTS TO CUBEWORLD

Kalpathi R. Subramanian
Assistant Professor
Department of Computer Science
The University of N. Carolina at Charlotte

Abstract Cubeworld is a 3D Virtual Reality (VR) application under development by the MITRE corporation. A stereoscopic display coupled with 3D input devices and a speech recognition system allows users to navigate and interact within 3D environments using an easy to use interface. Two enhancements were implemented to extend Cubeworld's capabilities, (a) viewpoint manipulation, which permits a user to look at the world from the vicinity of any object within the 3D world, and (2) a computer model of the Deployable Reconfigurable Command Center (DRCC) was imported into Cubeworld, to permit real-time navigation and inspection of the model, both as a complete model, or, in terms of its components.

ENHANCEMENTS TO CUBEWORLD

Kalpathi R. Subramanian

Introduction

The use of Virtual Reality (VR) and Virtual Environments (VE) in recent years has become a useful means to understanding and evaluating large amounts of information generated from a a variety of application domains. In conjunction with sophisticated devices for interaction, VR techniques can put users in immersive, non-immersive or augmented environments. Such environments have been used in applications such as medical imaging, education, human interface design, architectural building walkthroughs as well as in a number of military training and battlefield environments.

In this project, we have worked with Cubeworld, a particular VR application, that supports stereoscopic displays for output viewing and devices such as the VTi Cyber Glove or 3D Mouse for grabbing objects, Spaceball for navigation, and speech as an alternative means for communication and interaction. In particular, we have extended this application so that navigation can be rapidly switched to originate from within any of the objects within the environment. In essence, this involves moving the viewer to a new viewpoint, determined by an object selected by the user. In the second part of the project, a computer model of the *Deployable Reconfigurable Command and Control (DRCC)* has been imported into this application so that it can be viewed and manipulated within a stereoscopic virtual environment.

Overview of Cubeworld

Cubeworld [4] is an application under development by the MITRE corporation for navigating and interacting with 3D virtual environments. It supports stereoscopic displays, 3D input devices including the Spaceball (Silicon Graphics), the VTi Cyber Glove, as well as a Logitech 3D mouse. The application also uses BBN's HARK system, that allows the creation and manipulation of objects in the 3D world through audio commands. These multi-media capabilities allow a user to create or import 3D environments, select, group or highlight objects of interest, and navigate or fly through the environment with the help of the 3D input devices. A large collection of environments have been developed to demonstrate the relevance of these technologies to applications such as navigation of terrain models, with both land vehicles (tanks) and combat aircraft (B-52, F-22),

medical, office and work environments, etc. A major goal of the ADII (Advanced Displays and Intelligent Interfaces) is the evaluation of such technologies for their relevance and possible use in military applications, such as mission planning and rehearsal.

Cubeworld is implemented in C++, and currently consists of roughly sixty different classes (object types). It runs on Silicon Graphics workstations and exploits texture mapping extensively in its rendering modules for realism.

Cubeworld Main Loop

The main loop of Cubeworld consists of the following:

```
Initialize and create input devices
while (!quit)
{
        Increment clock
        Run all devices
        Run all objects
        Update all textures
        quit = check for input events
}
```

Thus, the main loop consists of "running" all of the objects, devices and textures. Objects are updated in terms of their position or orientation, devices are monitored for their new state (set via input events) and dynamic textures are generated for objects containing texture maps. At the end of the loop, the event list is checked for termination. All of the application behavior is hidden within its objects.

Objects, Devices and Coordinate Systems

A 3D environment consists of a list of *G_Objects*. Individual objects such as cubes, cylinders and more complex objects such as workstations, aircraft models (B52, F22) are all derived starting from a *G_Object*. Other objects such as Catalogs and Menu objects, which contain collections of objects are also derived from *G_Object*. They are distinguished by the coordinate system in which they are defined.

All input devices are part of the *G_Device* list. Devices have methods to retrieve position and state of their buttons, which are used to control the navigation. All objects and devices have a

run() method that involves updating the object position, retrieving the device status or location, as illustrated in the main loop above.

Four different coordinate systems are supported: (1) model (master) coordinates, that describe the geometry and attributes of the objects, (2) World Coordinates, the common coordinate systems for all objects (3) body coordinates, that are associated with the origin at the center of the screen (used by catalogs, menu objects and input devices), and, (4) screen coordinates, which are used for tracking head mounted display type devices, allowing the head to turn and look around. Homogeneous transformation matrices are used to move from one coordinate system to another: the transformation pipeline is as follows:

$$M = M_{position} * M_{location} * M_{viewpoint} * M_{viewmatrix}$$

where

$$\begin{aligned} M_{position} &= \text{Master to World coordinates} \\ M_{locaiton} &= \text{World to Body coordinates} \\ M_{viewpoint} &= \text{Body to Screen coordinates} \\ M_{position} &= \text{Screen to Perspective coordinates} \end{aligned}$$

Finally, depending on a particular environment, objects within that environment define a variety of behaviors and constraints. Thus, in the Hunter-Liggett tank simulation, the tank is constrained to travel over the terrain and follows the contours of the terrain, supports a scope and can fire weapons at enemy vehicles. In another example environment ("Terrain9"), a combat aircraft continually flies in a circle with a radar beam sweeping the terrain below and has the capability to identify ground vehicles. All of this is specified as part of the model that describes the object.

Viewpoint Manipulation

Navigation within Cubeworld is primarily through the Spaceball, a 3D input device. While this is appropriate for most situations, it was felt that it would be useful to be able to view the model from certain specific locations, for instance, from a flying aircraft. This is difficult to accomplish

using the Spaceball alone, especially if the object of interest is itself in motion. In other words, it involves placing a camera at selected "viewpoints" within the environment and be able to switch between them.

In order to implement this in a very general and automated fashion, it was decided to locate potential viewpoints in the vicinity of the objects. Once an object is specified by the user, the application simply switches the viewer to a point close to the object. Additionally, the user may use the Spaceball to view the environment from any orientation, while being anchored to this viewpoint. In case the object is itself moving at a constant velocity, the "viewpoint" also moves with the object, thus enabling the user to view the environment from a flying aircraft, for instance.

Implementation

A new class, named *ViewpointManipulator* (detailed in the Appendix) was implemented to effect viewpoint manipulation within Cubeworld. A menu system facilitates interactive object selection. Additionally, Cubeworld could now be operated either in VIEWPOINT mode, when viewpoint selection can be performed, or, in NORMALVIEW mode, when the 3D locator devices are used for navigation. In VIEWPOINT mode, the user is allowed to arbitrarily orient the camera about the viewpoint, but translation is not permitted. This decision was primarily made to facilitate viewpoint location within moving objects.

The *ViewpointManipulator* object contains methods to switch between the two viewing modes, determine the bounding volume extent of an object that has been selected for viewpoint location, a "run" method, which computes the affine transformation to locate the viewer, as a function of the Spaceball input parameters and the prior transformations [2]

A second class, *MB_ViewpointManipulator* implements a simple menu system for object selection. It contains methods to activate and deactivate the menu, scroll up or down the menu (either 1 item at a time or page up/down). By default, the menu is created with the objects in the current environment. However, within cubeworld objects can be dynamically created by the user or generated over the network. To include these objects within the viewpoint selection menu, a "Refresh" item is provided that updates the object names in the menu. All of the menu operations are controlled using the buttons on the Spaceball. The Spaceball has an eight button keypad and an additional button on the sensor ball. Button bindings are as follows:

Button No.	Action
8	Toggle between VIEWPOINT and NORMALVIEW Modes (Viewpoint Menu Inactive)
4	Menu On/Off
5	Scroll Up 1 item
6	Scroll Down 1 item
7	Page Down
8	Page Up (Viewpoint Menu Active)
0	Select Item

The operation of the Viewpoint mode is as follows. Cubeworld, by default starts in NORMALVIEW mode. In this mode, navigation is via the Spaceball. It outputs six values, three translation factors and three rotation angles, corresponding to the three principal axes. These values are used in moving and orienting the viewer at each clock instant, followed by drawing the objects in the environment. This operation is equivalent to moving and rotating the viewer with six degrees of freedom.

Pressing button 8 switches Cubeworld into VIEWPOINT mode; if no object has been selected (which is the case initially), the viewpoint menu is posted. The button bindings described above are used in selecting an object. Note that button 0 is on the sensor ball. Once an object is selected, the corresponding object's bounding extent is calculated and the viewer is positioned at the center of this extent. Currently a simple rectangular extent is calculated from the vertices that make up the object. From this point on, only the Spaceball's rotation values are utilized. Thus the navigator can "look around" the environment about the viewpoint, but not move in any direction. This decision was made primarily to handle the case when the object itself could be in constant motion, such as a flying aircraft or land vehicle. In these cases, the user "moves" with the object. To switch to a different object, the menu is posted (using button 4) and a new object selected. Finally, the Refresh item in the menu will update the menu, by adding new objects created during the session.

Incorporation of DRCC

The *Deployable Reconfigurable Command and Control (DRCC)* is a project currently underway at the Air Force Research Laboratory directed towards rapid deployment of command centers for use in the battlefield. Central to this project is the use of the *Interactive Data Wall* [3], which

consists of a high resolution wall display with various wireless input devices shared among multiple users, including speech interaction, and camera tracked laser pointers. The primary goal of the DataWall is the ability to bring in large amounts of information for analysis by multiple users. Making this part of the DRCC fits within a battlefield environment where data from different parts of the battlefield might be sent to the command center for evaluation and rapid response. The DRCC is also being designed with components that make it highly mobile and to be set up or dismantled within a short period of time.

In this part of the project, a computer model of the DRCC was designed by Jeff Baumes [1], using the Designers WorkBench modeling toolkit. A computer model is useful in previewing the center prior to actual construction. As Cubeworld allows such models to be displayed in stereoscopic 3D and capabilities to pick individual objects with the Cyber Glove, it was felt useful to import this model into the system. Accordingly, the DRCC model was imported into Cubeworld (in Flight format). While this allows the entire model to be viewed, it was felt that it would also be useful to access and manipulate individual components of the model. Accordingly, the DRCC model was partitioned into 3 major components, (1) DRCC Inside, (2) DRCC Outside, and (3) DRCC DataWall screen. In the current implementation (which is not quite complete), it is possible to selectively display any of the three components.

Implementation

Cubeworld supports catalogs, which are groups of objects, assembled into panels. Catalogs are an intuitive means to creating and manipulating the objects within an environment, and function somewhat analogous to a 3D interactive editor. Cubeworld allows objects to be created from a catalog by simply grabbing them from the catalog using the Data Glove and locating them within the 3D environment. Catalogs can have multiple pages, with up to six objects per page.

A catalog was created containing all of the components that make up the DRCC model. The model consists of the following components:

1. Mobilizer Front
2. Mobilizer Back
3. Power Hub

4. Shelter Front
5. Computer Rack
6. UPS (power supply)
7. Chair
8. Mouse
9. Keyboard
10. Monitor
11. Sun Monitor
12. Projector
13. Sun Ultra10 (Computer)
14. Sun Ultra60 (Computer)
15. Wireless Microphone

In addition to these components, four additional components were incorporated for convenience, (1) the full DRCC model, (2) DRCC Inside, which contains the objects within the interior of the center, (3) DRCC Outside, which contains components such as the Shelter, Tactical Quiet Generator Set, and Environmental Control Unit, and, (4) DataWall screen. All components are imported into Cubeworld as Flight models.

Finally, the three partitioned components of the DRCC model can be selectively turned On or Off, and is controlled by the Spaceball buttons 5, 6 and 7. A boolean flag within the DRCC object class determines if the particular component needs to be drawn or not.

Conclusions

Two important extensions to Cubeworld have been implemented and tested. While the implementation has been well tested, additional enhancements in viewpoint manipulation are possible. For instance, in the current implementation, viewpoints are restricted to be in the center of the

bounding extent. This is a problem if the object is completely solid or enclosed on all sides. Small amounts of translation from the current viewpoint (say, to the edges of the bounding extent) would address this problem.

Additionally, not all objects in the example environments support bounding extents. Some objects (Window Assembly in "room9" environment) are simply object collections. In these cases, it is necessary to examine the object and if necessary, derive a bounding extent of its constituent objects.

Acknowledgements

Thanks to AFOSR and the Rome Research site (in particular, Dr. Richard Slavinski) for sponsoring this project. My thanks also to Pete Jedrysik for his assistance in testing the implementation and for several valuable suggestions that improved the implementation. Mike Wingfield was very helpful in answering numerous questions on his implementation of Cubeworld.

REFERENCES

[1] J. Baumes. DRCC Model, August 1998. Personal Communication.

[2] J.D. Foley, A.Van Dam, S.Feiner, and J.F.Hughes. *Computer Graphics: Principles and Practice.* Addision Wesley Publishing Company, Reading, Massachusets, 1992.

[3] J. Lipa. DataWall, August 1998. www.rl.af.mil/programs/ADII/adii_main.html.

[4] M. Wingfield. Cubeworld Implementation, August 1998. Personal Communication.

Appendix: Class Descriptions

We describe the major classes that have been added to Cubeworld, as part of this project.

1. ViewpointManipulator

Embeds data types and methods to switch viewpoints between different objects in the current environment.

```
class ViewpointManipulator
{
        friend  VpmMenu;

        private:
                Matrix3D m, m_rot;
                Vector3D delta_twist;
                float angle;
                static  Viewmode current_mode;
                static  ViewpointManipulator *current_vpm;
                static  MB_ViewpointManipulator *current_mb;
                static  G_Object *current_object;
                static  Vector3D current_location;
                static void getExtent (G_Object*, Vector3D&, Vector3D&);

        public:
                ViewpointManipulator(VString& scene_name);
                ~ViewpointManipulator();
                void run(void);
                G_Object* getCurrentObject (void) { return current_object; };
                static void setCurrentObject (G_Object *g);
                static void setCurrentMode (Viewmode mode) { current_mode = mode; };
                static  Viewmode getCurrentMode (void) { return current_mode; };
                static void switchMode (void);
                static  ViewpointManipulator* getViewpointManipulator(void);
                static  MB_ViewpointManipulator* getMenuBar (void) { return current_mb; };
                static void setCurrentViewpointManipulator(ViewpointManipulator*) { current_vpm = vpm
                void setSpaceballParams (Vector3D twist, float a) { delta_twist = twist; angle = a;};
                Vector3D getOrigin (void) { return current_location; };
};
```

2. MB_ViewpointManipulator

This class contains the methods to implement the menu used to switch between viewpoints.

```
class MB_ViewpointManipulator : public G_Object
{
        private:
                int active;
                VpmMenu *vpm_menu;
                Vector3D loc_sc;
                VString scene_name;

        public:
                MB_ViewpointManipulator(VString&);
                 MB_ViewpointManipulator();
                int Active(void) { return active; } ;
                VpmMenu *getMenu (void)  return vpm_menu; ;
                void setMenu (VpmMenu* menu) { vpm_menu = menu; } ;
                void Activate (void);
                void DeActivate (void);
                VString getCurrentSceneName() { return scene_name; };
                void run (void);
                void draw(const int);
};
```

3. DRCC Classes

```
class DRCC_Inside : public FltModel
{
        private:
                int visibility;
        public:
                DRCC_Inside (const char*, const char*);
                void draw (const int);
                void run (void);
                void setVisibility (int v) { visibility = v; };
};


class DRCC_Outside : public FltModel
{
        private:
```

```
        int visibility;
    public:
        DRCC_Outside (const char*, const char*);
        void draw (const int);
        void run (void);
        void setVisibility (int v) { visibility = v; };
};


class DRCC_DataWall : public FltModel
{
    private:
        int visibility;
    public:
        DRCC_DataWall (const char*, const char*);
        void draw (const int);
        void run (void);
        void setVisibility (int v) { visibility = v; };
};
```

A DISTRIBUTED CONCURRENT INTRUSION DETECTION SCHEME BASED ON ASSERTIONS

Shambhu J. Upadhyaya
Associate Professor
Department of Computer Science and Engineering

State University of New York at Buffalo
Buffalo, NY 14260

Final Report for:
Summer Faculty Research Program
Air Force Research Laboratory
Information Directorate
Rome Research Site

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Air Force Research Laboratory
Information Directorate
Rome Research Site

September 1998

A DISTRIBUTED CONCURRENT INTRUSION DETECTION SCHEME BASED ON ASSERTIONS

Shambhu J. Upadhyaya
Associate Professor
Department of Computer Science and Engineering
State University of New York at Buffalo

Abstract

This document describes a new intrusion detection scheme based on concurrent monitoring of user operations. In this scheme, prior to starting a session on a computer, an auxiliary process called watchdog first queries users for a scope file and then generates a table called a sprint-plan. The sprint-plan is composed of carefully derived assertions that can be used as a basis for concurrent monitoring of user commands. The plan is general enough to allow a normal user to perform his task without much interference from the watchdog or system administrator and is specific enough to detect intrusions, both external and internal. A distributed watchdog process architecture based on the notion of verifiable assertions is presented. This scheme is a significant enhancement over the traditional approaches that rely on audit trail analysis in that the intrusion detection latency is much shorter and can help fast recovery from successful break-ins. The issues related to recovery following the on-line detection of intrusions are also discussed.

A DISTRIBUTED CONCURRENT INTRUSION DETECTION SCHEME BASED ON ASSERTIONS

Shambhu J. Upadhyaya

1 Introduction

Information survivability in the wake of abuse, misuse and malicious attacks is an important issue in today's distributed systems. It is necessary to develop countermeasures to deal with the attacks or intrusions since some of the attacks may succeed despite vigorous information security measures. It is almost impossible to close all security loopholes and guard against malicious break-ins as well as abuse of systems by legitimate users [1]. One of the countermeasures is to detect intrusion immediately and initiate recovery procedures to undo the damage.

A variety of intrusion detection techniques and tools exist in the computer security community. Though these techniques follow different approaches for intrusion detection, audit trail analysis has been used as the last line of defense [1]. In these methods, the user behavior is monitored for certain patterns of abuse by looking at the audit data. Unfortunately, intrusion detection schemes based on audit trail analysis do not offer much in terms of damage containment because these approaches are passive, after-the-fact solutions [2]. They are known to be largely firsthand and heuristic. In order to contain the damage effectively it is essential to detect intrusions concurrently, that is, while the intrusion is in progress. Detection with low latency can effectively contain information contamination in a distributed system. This, in turn, can help speedy recovery and restoration of service. The development of such a concurrent monitoring scheme with the objective of detecting intrusions with low latency is the main focus of this research.

In order to develop concurrent monitoring of any operation and flag an exception, one must have a flow graph of the intended operations. An analogy exists in the traditional fault tolerance area where concurrent error detection has been successfully implemented at instruction level using the control flow of user application programs [3]. However, intrusion detection is different from this because intrusions may not change instruction-level control flow. Instead, one must look at a higher level such as the user command level where intrusions can be better observed. However, control flow of user operations is not known *a priori*, making concurrent monitoring a difficult task. As an alternative, we devise a reference table to compare the user operations with.

The proposed technique of concurrent intrusion detection starts by first generating a plan for the potential user and then monitoring the user for any significant deviation from this plan. This plan is composed of verifiable *assertions* that are automatically generated on the basis of a few system usage inputs provided at the beginning of a session. Once an assertable

plan is generated, the user will be monitored to see how close the plan is maintained during a given session. Certain tolerance limits can be set to take care of unplanned, but authentic deviations. Any significant deviations from the plan can be construed as an intrusion, external or internal. At this stage, either an interactive dialog is opened with the user or the information is shared with other monitors in the network to minimize false alarms and to initiate system level recovery if the threat is viewed real.

Our approach is based on sound principles that have been successfully used in concurrent monitoring of processor operation. In addition, the new approach offers several advantages. First of all, no audit trail analysis is needed. Audit trails are generally huge and the analysis and filtering out of useful information is expensive [1]. The assertable plan can take into account user dynamics and any intended changes in the user behavior. Therefore, no special learning is needed as in rule-based systems. Both inside abuse and external intrusion can be minimized. Since our scheme does concurrent monitoring with low latency of detection, it can be combined with known recovery schemes for effective damage containment and recovery.

Some background and related work are given in Section 2. An overview of the methodology and an architecture of distributed concurrent intrusion detection appear in Section 3. The algorithm and illustrations are given in Section 4. Section 5 combines low latency detection and recovery issues. Conclusions, comparison to related work and future directions are given in Section 6.

2 Background and Related Work

Intrusion detection which is concerned with the detection of unauthorized access of computer systems is a critical phase in information survivability [4]. A model for intrusion detection was first given by Denning [5] which uses audit trail analysis as a basis for detection. The detection approaches based on audit data analysis use various techniques such as rule-based expert systems, model based systems and state transition analysis. A survey of these traditional intrusion detection techniques is found in [1].

Feldman et al. [2] observe that the current intrusion detection technology focus is largely *after-the-fact.* What is needed is a proactive/predictive intrusion detection technology incorporating advances in pattern matching, user profile analysis, and intrusion signature recognition. The ultimate goal is the ability to anticipate attacks before they occur. To this end, there are already several efforts in progress. Keystroke monitoring, command-line auditing, tripwires, signature analysis, real-time pattern matching and distributed network monitoring are some of the examples. The technique proposed in this document differs from these approaches in that our effort supports the building of self-aware hardware and software systems that adapt, in real-time, to external and internal anomalies. Before presenting the details of our scheme, we briefly review other real-time intrusion detection schemes found in

the literature.

Clyde [6] suggests *keystroke* monitoring as a means to dynamic intrusion detection. This is a behavior pattern monitoring which can be done on-line unlike audit trail analysis which is essentially after-the-fact. The idea is that the intruder will have a different keystroke pattern compared to the legitimate user's. This technique has been discounted as impractical by Kumar and Spafford [7] due to the myriad ways of expressing the same attack at the keystroke level. Aliasing has also been cited as a reason for the defeat of this technique. Command-line auditing has been tried on Unix systems by several researchers as a means to intrusion detection. Unix commands such as 'lastcomm' or 'history' can be used for this purpose. This technique is ad hoc and is similar to keystroke monitoring. No concrete work has been found in this direction.

USTAT is a Unix-based real-time intrusion detection tool developed by Ilgun et al. [8]. It uses a state transition analysis where penetration is identified as a sequence of state changes from a known initial state to a final compromised state. The technique is audit trail-based. It works on the data collected by the C2 basic security module of SunOS and keeps track of only those critical actions that must occur for the successful completion of penetrations. It differs from other rule-based intrusion detection tools that rely on pattern matching of sequences of audit records.

The work of Kumar and Spafford [7] uses a novel pattern matching scheme for real-time intrusion detection. A specialized graph called colored Petri Net is used to represent the knowledge about attacks in the form of signatures in the Petri Net guards. The vertices of the net represent system states. This model provides the ability to specify partial orders and subsumes matching of sequences and regular expressions. Generality, portability and flexibility have been cited as the main benefits of this model.

Several real-time network monitoring systems have been developed over the past years. NADIR [9] is a network anomaly detector and intrusion report tool which was originally designed to accept audit logs from a Los Alamos network security controller running a homegrown version of Kerberos (the original version of Kerberos was developed at MIT as a cryptography-based network authentication protocol). A more powerful version of NADIR called UNICORN accepts audit logs from Cray Unix called UNICOS, Kerberos and common file systems and analyzes them and tries to detect intruders in real-time. DIDS [10] is a distributed intrusion detection system which looks at and correlates the connections on multiple machines to the initial login. This allows the actions of one person hopping among machines and multiple name changes to be associated to his initial connection into the monitored network. As the information is generated, an expert system evaluates the information and determines if some action is required.

More recently, the computer science lab at SRI International has undertaken a project called EMERALD (Event Monitoring Enabling Response to Anomalous Live Disturbances) [11]. This project is developing a distributed monitoring scheme which uses a combination

of a signature engine and a profiler engine within the monitor for intrusion detection. The approach is hierarchical in the sense that the individual service monitors coordinate the detection with a network monitor in order to provide a global detection and response capability to the network. The signature analysis is a process whereby an event stream is mapped against abstract representations of events that indicate undesirable activity.

The research group at Purdue is developing an adaptive network monitoring scheme using autonomous agents. Their approach [12] is distributed in the sense that one agent is used per node instead of a monolithic entity. The agent is somewhat similar to the monitor of Emerald [11]. This architecture also uses a hierarchical approach like Emerald.

3 Methodology Overview and System Architecture

The previous section reviewed some of the real-time intrusion detection schemes. Most of these schemes are based on audit-trail analysis and pattern matching. Their real-time notion stems from the fact that they use either efficient pattern matching techniques or employ distributed decision making to speed up detection. We approach the real-time intrusion detection problem from a different point of view, namely, tconcurrent, on-line monitoring. This approach essentially does away with the passive audit trail analysis. The model and the simplifying assumptions are given next.

3.1 The Model and Assumptions

In our definition of a distributed system, we include a network of computers which service users on the basis of an account and a password. Further, we assume that the users on different machines have the same userID although the passwords could be distinct. This model precludes the monitoring of web surfing and anonymous ftp activities. No specific topology is assumed for the network. All the communications between nodes is by message passing.

The above model makes our intrusion detection approach unique in that all intrusions are abstracted as happening through well-defined user sessions which are invoked through userID and password submission. This leads to the following observations.

1. The intrusions are of only two types: (a) external intrusions caused by logging into a legitimate user's account by way of cracking password or by joining an open session left logged on by a legitimate user; (b) internal misuse occurring by abusing the privileges or by forcing on others' accounts and manipulating files with a malicious intent.

2. The problem of intrusion detection simply transforms into monitoring the well-defined user sessions. This is because whatever be the source, the intrusion will be through one of the well-defined user sessions. No distinction exists between external or internal

intrusions as far as the detection is concerned. Thus, external intrusions and internal abuse can be handled in a unified way.

We also assume that a user session on a node is of finite length. This facilitates consideration of temporal information for focused monitoring of user operations. Some terminologies are defined next.

3.2 Definitions

Definition 1 *An* **intrusion** *is defined as any deviation from a predetermined operational behavior.*

This definition of intrusion includes such activities as masquerading, legitimate user penetration, and legitimate user leakage [5]. Other intrusions of the type Virus, Trojan horse and denial-of-service are not considered here. We assume that existing security measures on the system can detect most such anomalies. However, the userID-password abstraction in our model may allow the detection of many cases of intrusion before leading up to anomalies such as denial-of-service.

Definition 2 *A* **watchdog** *is a process that concurrently monitors user commands typed in from a keyboard or submitted in the form of a script or a macro.*

A watchdog process is spawned immediately following a userID and password submission and the creation of a user session. Only one process per userID is created.

Definition 3 Session-Scope *is a file containing a list of intended activities submitted by a user at the beginning of a session.*

This file is a high level description of user operations in a prescribed format. This is analogous to a *flight plan* generally submitted to the air traffic controller by a pilot before flying his private plane. Since only one monitor session is set up per user account, only one session-scope is accepted per user account even if multiple sessions are opened on a given user account. The session-scope, once submitted, is treated as a secure document and is not accessible by anyone. As will be seen later, having one session per userID over a given period of time facilitates easy monitoring for intrusion.

Definition 4 *A* **verifiable assertion** *is a quadruple which associates a user with the intended operation over a given period of time. The format of the assertions is as follows.*

$$(subject,\ action,\ object,\ period) \tag{1}$$

where 'subject' is a user (along with additional IDs such as terminal identification, IP address etc.), 'action' is an operation performed by the subject, such as login, logout, read, execute, and 'object' is a receptor of actions such as files, programs, messages, records, terminals, printers etc.

These verifiable assertions are generated in advance for each user event specified in the session-scope file. The terminologies used here are similar to the audit trail terminologies [5], but certain entries such as *resource-usage* and *time-stamp* found in the audit records are not meaningful here. Instead, we associate a temporal characteristic called *period* which is an additional parameter signifying the time interval for the usage of a given user command.

Definition 5 *A* **sprint-plan** *is a collection of verifiable assertions.*

This is a table automatically generated as a response to a user's session-scope file. The sprint-plan can also be viewed as a signatured stream of commands that is generated, in no particular order, for the purpose of on-line monitoring.

3.3 Basic Principle

Our technique of intrusion detection using verifiable assertions is similar to the notion of control flow checking in fault tolerance [3], [13], [14]. In control flow checking, a preanalysis of the program is done to generate a control flow graph of the application. Signatures or assertions are embedded into the instruction stream at compile time to generate a reference graph. At runtime, the execution is monitored and at designated intervals, the runtime signatures are compared with predetermined signatures of the reference graph. Any discrepancy between the actual signature and the expected signature indicates an error. Both instruction level bit errors and control flow errors are detected by this scheme.

Though the control flow checking concept can be extended to intrusion detection, instruction level models are not applicable here because intrusions may not change instruction-level control flow. Nonetheless, monitoring at the file or directory level of granularity may be sufficient for intrusion detection [5]. Accordingly, we use a different approach for the derivation of a reference graph as described below.

The user starts a session on a computer in a standard way, that is, by logging in. The system then queries the user for a session-scope. The user gives a summary of his intended system usage. An example of a session-scope is as follows. A user plans to work until 5 pm. He will be working on a research paper. He will perform email and web browsing. He will run some simulation on a particular server. He will clean up his mail folders and may do some other miscellaneous items if he has time before the end of the day. The purpose of such a plan file is to get a basic understanding of what the user's intention is. It does not necessarily mean that this user must terminate his session at 5 pm or he has to strictly follow the original plan. Enough flexibility will be built into the system so that the user does not constantly have to interact with the monitor. Once the scope file is submitted, the user is allowed to continue with his session. Meanwhile the system translates the scope file into a sprint-plan. Since no ordering of events is possible on the activities of the user, the sprint-plan is simply a table of verifiable assertions. It has no control flow information as such.

The assertions give a mechanism for monitoring the user behavior. These assertions are generated automatically by parsing the scope file and interpreting the user inputs properly. An important component of the assertion is the subject field. The subject field is generated from the userID and other unique identifications such as the IP address of the workstation, tty number of the terminal being used etc. All such information will be coded into the subject field. For instance, a user may wish to open multiple login sessions. As long as such intent is expressed in the scope file, a more general subject coding can be done for this user in order to allow him to work from different terminals or set up multiple login sessions. As noted earlier, there is only one watchdog process per user per system even though multiple sessions are opened.

When the user is in session, the watchdog monitors the user commands and checks if the command is the one he originally intended to execute. Any significant deviation from the plan is an indication of potential intrusion. Several techniques are used to minimize false alarms and to build robustness to the monitoring scheme. These details will be discussed later. The intrusion detection architecture is given next.

3.4 The Architecture

Our intrusion detection model is amenable to hierarchical monitoring where the lowest level of hierarchy is the user-session level monitoring. Hierarchical arrangement of watchdog monitors is highly effective in distributed systems as evident from other intrusion detection schemes such as Coast [12] and Emerald [11]. Figure 1 shows the overall architecture for a network of computers consisting of N nodes and a file server.

A watchdog process is set up for each user on a given node. However, the process remains dormant until a user starts a session on the node. These watchdogs are essentially instances of the same process, monitoring the various user sessions. They remain restricted to the local nodes, but once operational, interact with a master watchdog which is responsible for coordinating distributed system monitoring.

In addition to the user watchdogs and master watchdogs, each local network has a separate watchdog called a File Watchdog. The function of the file watchdog is to monitor accesses to secure files on the file server as shown in the figure. Such a separate watchdog with the knowledge of the secure files and any privileged information will be more effective in detecting unauthorized disk accesses to secure data than incorporating the detection functions into the user watchdog. Since secure data may not reside on the disks of individual nodes, a central watchdog guarding the file server will be easier to implement than a distributed implementation. The file watchdog will interact with the master watchdog on the individual nodes to coordinate the dissemination of intrusion detection and to initiate recovery. The architecture of the individual user watchdog is shown in Figure 2.

The watchdog process receives input from the User Command Buffer and/or the Operating System. The Atomic Operation generator converts user command lines into an on-line

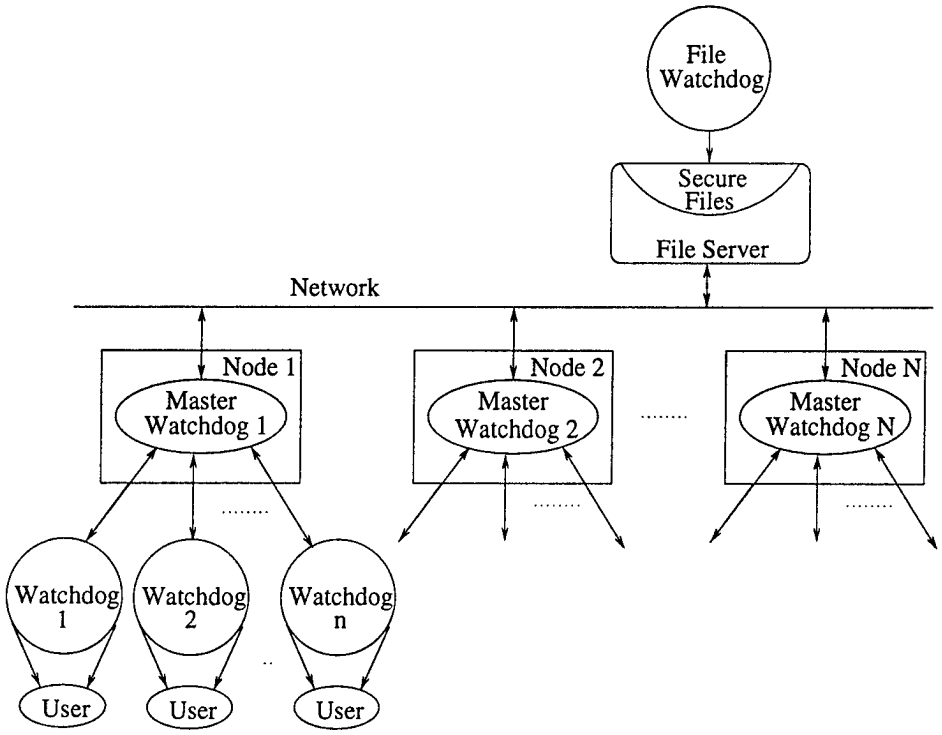


Figure 1: Overall Architecture of a Distributed Intrusion Detection Scheme

assertion statement. The Inclusion Checker module verifies if the assertion statement generated on-line is in the set of previously generated verifiable assertions. This verification can be done by a simple table lookup search. The synchronization of watchdog and user operations is not an issue here because the elapsed time between successive user commands or the converted atomic operations is expected to be much higher than the time needed for the watchdog to complete the verifications.

The watchdog has a dialog box which, with the help of a Counter, initiates a dialog with the user if the user is found to deviate significantly from the original plan. Based on the user response received at the Dialog Initiator, a decision is made to issue an exception to the master watchdog. An Intrusion Signal is issued by the master watchdog after carefully processing the monitored data.

4 Algorithms and Illustrations

The preparation of an accurate sprint-plan at the beginning of a session could be a quite difficult process. If the session-scope file is too cryptic or imprecise, it is possible to interpret legitimate use of the computer as intrusion, giving rise to false alarms. An important user requirement is the availability of enough flexibility to work on a session on a user's own workstation. This adds some burden on the sprint-plan generator. The generated sprint-

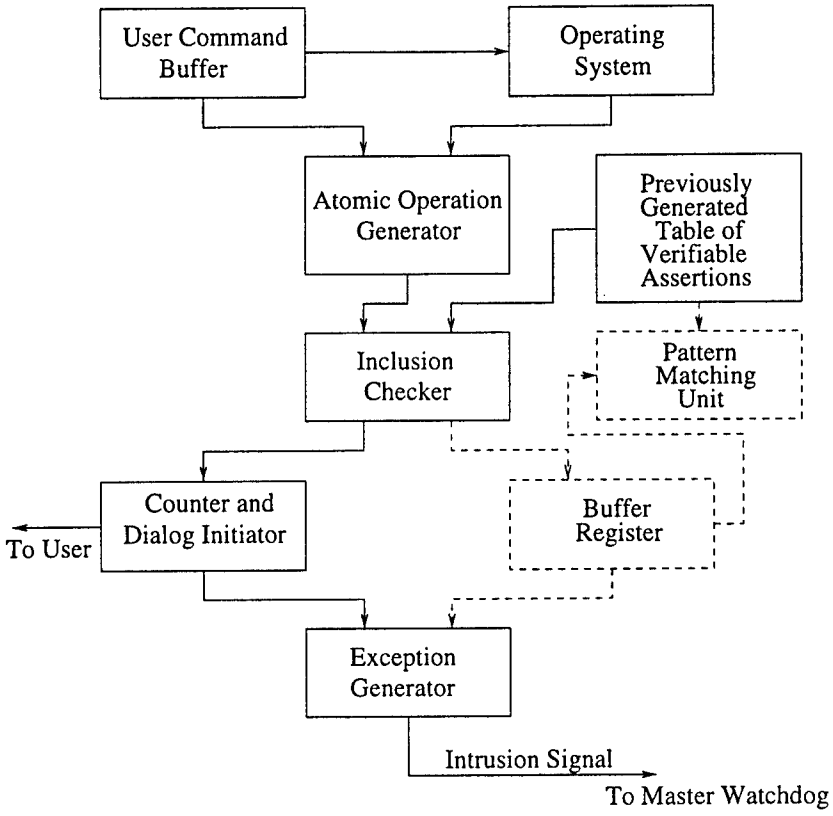


Figure 2: Block Schematic of the Watchdog Process

plan must be detailed enough to give flexibility to the user but at the same time should not be too exhaustive to the extent that the table of assertions is a superset of all possible user commands. Certain techniques such as accounting checks or reasonableness checks [15] can be used to aid in the translation of user inputs to a sprint-plan. A good graphical front-end with automated features for data entry will also be useful.

The concurrent intrusion detection scheme has two phases of operation. The first phase is an initialization phase which solicits input from the user and generates a sprint-plan as described above. We do not delve into this in this document. The second phase is the runtime phase during which on-line monitoring is performed. The algorithm for the second phase is given below.

4.1 The Algorithm

A basic algorithm is given first. Certain tolerance limits are used to allow for user deviations from the intended plan. Appropriate counters are used to keep track of such deviations.

Algorithm:

1. Set *monitor_rate, tolerance_limit, counter*;
2. For all *user_command_line* do
3. Decode *user_command_line* into atomic operations;
4. If each atomic_operation in *sprint_plan* then
   (a) no_error, go to Step 3;
5. else
   (a) If *subject_ID_violation* then
      i. Set *intrusion_signal*, exit;
   (b) else
      i. *counter++*; /* increase count on permissible unplanned commands */
      ii. If *counter* $\geq$ *tolerance_limit* then
         A. If *provision_for_future_changes* in *session_scope* then
         B. Reset *counter*, go to Step 3;
         C. else
         D. Issue message to user to update *session_scope*;
         E. If *user_response* YES then
         F. Compare new *session_scope* with original *session_scope*;
         G. If *criteria* not met then /* See explanation below */
         H. Issue *intrusion_signal*, exit;
         I. else
         J. Reset *counter*, go to Step 3;
         K. else
         L. Issue Intrusion_signal, exit;
      iii. else
         A. go to Step 3;

One can choose to monitor every single user command or set a particular monitoring rate (Step 1) depending upon parameters like system load and specified security levels. The session-scope to be submitted at the beginning of a session need not be complete. The user can update his plan at a later time. The watchdog will query the user for an update based on a tolerance limit imposed on the amount of deviation from the original plan.

The user input line in the form of a singular command, command alias, macro or a script is converted into a set of atomic operations, irrespective of whether it is interactive or a batch submission. These atomic operations will have the subject ID resolved based on runtime characteristics. If the atomic operations are found in the sprint-plan previously generated (Step 4), the session continues without interruption. If there is a mismatch, further examination is conducted to see if there is a subject ID violation (Step 5.a). If so, an intrusion is obvious and appropriate signals will be sent to the master watchdog. Otherwise, the mismatch is interpreted simply as a deviation from the original plan and the count on permissible unplanned commands is incremented (Step 5.b.i). If this count reaches a tolerance limit (Step 5.b.ii), a check is made to see if the user has made provisions in the session-scope file for future updates. If no such intent was specified in the session-scope file, a message is issued to the user to indicate that the original plan is too coarse and an update is in order. If the user responds affirmatively (Step 5.b.ii.E), he is expected to submit a revised session-scope. The revised session scope is examined for certain criteria. If the criteria are met, the counter is reset and the session continues without further query. The updated session-scope is translated to a new sprint-plan which will be used for the rest of the session or until the sprint-plan is updated again. If the criteria are not met or the user does not update his session-scope, an intrusion is flagged and appropriate action is initiated.

At Step 5.b.ii.F, a mere submission of an updated session-scope cannot be considered as a benign act. It is possible for a crafty intruder to perform only incremental changes or to include a very general plan in his updated session-scope and evade detection. Therefore, the updated session-scope must be examined for certain criteria. A comparison between the new file and the old file may reveal some of the user (intruder) intentions. One must also make sure that the updated scope file does indeed include the activities which triggered the counter overflow. Such checking will prevent intruders from continuing the intrusive activities by hiding the operations that triggered the counter overflow. Several such simple and effective criteria can be developed using reasonableness checks but this discussion is outside the scope of this document.

4.2 Intrusion Scenarios and Some Enhancements

The algorithm presented above offers much flexibility and provision to the user to update his plan so that false alarms are minimized. The algorithm is also capable of detection of many intrusion scenarios. Consider the situation where two logins are made on the same account. Multiple logins is possible due to genuine use by the user or due to an intrusion while a legitimate user is in session. There are four distinct cases to consider.

- Case 1: Both logins are legitimate. This is the most common scenario. For instance, a user starts a session from his desktop and after a while goes to his laboratory and remotely logs on to his account from a terminal or workstation in the lab.

- Case 2: The first login is from the legitimate user and the second login is from an intruder.
- Case 3: The first login is from the intruder and the second login is from the legitimate user.
- Case 4: Both logins correspond to intrusions.

For all the four cases, irrespective of the number of logins, our model treats the usage of the account as a single session. So, there is only one session-scope file during the entire session.

4.2.1 Resolving the Cases

Case 1. In this case, the user is expected to include the intent of multiple logins in the session-scope file and hence, the session continues without interruption. If no request is made initially, remote logins or multiple logins from the same terminal is prohibited as a security measure.

Case 2. This case is handled easily if the user does not indicate the desire for multiple logins in his session-scope file. If the user admits multiple logins in his session-scope and the intruder is able to login from a prescribed terminal or workstation, the break-in succeeds. However, if the intruder is oblivious of the watchdog, he is likely to deviate from the activities given in the session-scope file and detection will be imminent. Even if the intruder is aware of the monitor watchdog, his activity is likely to raise an exception since he will not know what is contained in the original session-scope file. Note that only the originator of the session can update the session-scope file.

There is some inherent security built into the algorithm if Case 2 occurs. If an intrusion occurs while the legitimate user is in session, the user will be contacted by the watchdog whenever the counter overflows (Step 5.b.ii). If the user believes that he has stayed within the session-scope for most part, he can choose not to update his session-scope. Too many requests from the watchdog to update session-scope is a clear indiction that someone else has intruded the system.

Case 3. Case 3 occurs when the masquerader who logs in first admits multiple logins in his session-scope file. If the masquerader does not allow multiple logins to begin with, the legitimate user will be denied access when starting system usage. He is then expected to contact the system administrator. This will eventually result in intrusion detection. On the other hand, if multiple logins are admitted and if the legitimate user does not log in during the intruder's session, the intrusion continues to be in place and there is no way of detecting this aliasing problem using our algorithm. In such cases, we need to rely on other techniques such as recognition of unusual patterns in the user behavior. In the case where the intruder admits multiple logins and the legitimate user logs in while intrusion is in progress, the

absence of a query from the watchdog to the user for a session-scope file may raise suspicion. The user, cognizant of the watchdog monitoring, is made to believe that someone is using his account before he has logged on or that the watchdog monitor itself is turned off. Even if the user takes no action (or say, is not cognizant of the watchdog), the continuation of his access will have a high probability of resulting in a significant deviation from what is in the masquerader's session-scope. The algorithm has the capability to capture such deviations.

Case 4. Case 4 can occur if the intruder himself initiates multiple logins or the second intrusion occurs from a different individual while the first intrusion is in progress. We assume that the probability of either of the events is small. However, when this happens, the latter scenario is identical to Case 3 if the second intruder is viewed as a user rather than as an intruder. In this case, the interactions of the events generated by the intruder and the user will eventually result in the detection. If the former situation occurs, the detection is not likely to happen by our algorithm and we need to rely on other techniques such as recognition of unusual patterns in the user behavior.

4.2.2 Enhancements

Several refinements to the basic algorithm are possible to enhance the robustness of intrusion detection.

Monitoring Sequence of Operations. In Step 5.b.ii.F, certain criteria are used for discriminating intrusion from normal use. An enhancement can be added by buffering the on-line assertions and comparing sequences of assertions with some predetermined patterns indicative of possible abuse. This enhancement is shown in Figure 2 by dotted lines. Also, one could look at the frequency of certain operations or other temporal characteristics of system usage to enhance the detection.

Voluntary Input of Updates. In the current version of the algorithm, the user is queried to update the session-scope (Step 5.b.ii.A) if there are significant deviations from the original plan. On the other hand, the user himself can voluntarily submit changes to his original plan on a need basis. Of course, if the scope file is modified too many times or too frequently, the user may be targeted as a possible suspect.

On-the-fly Admittance of Multiple Logins. Currently, if the user does not admit multiple logins in his session-scope file an intrusion is flagged when the second login is attempted. Flexibility can be added by querying the user to update the session-scope file on-the-fly to admit multiple logins. This way, the subsequent logins can succeed without terminating the original session.

Two Level Counters. Another enhancement is possible in Step 5.b.ii.B. In this step, currently, the counter is simply reset to allow for anticipated deviations when the tolerance limit reaches. If there are significant deviations, the counter may reach the limit and get reset a few times without further scrutiny. When counter reset occurs repeatedly, a second counter can be used to permit only a certain number of counter overflows. This hierarchy of

counters will add further security to the detection mechanism.

4.3 An Illustration

The session-scope file for a typical user is given below in a tabular form. More sophisticated graphical user interface can be built to ease the user input procedure.

query		User Input
Node name?	→	robin.eng.buffalo.edu
Userid?	→	shambhu
Expected session duration?	→	8am - 5pm
Multiple logins?	→	yes
From which terminals?	→	In Room No. Bell 600
Type of activity with time preference	→	
Network related:	→	ftp, rlogin
Local activities	→	LaTeX job on a research paper ()
	→	seqsim simulation
		on beatrix.eng.buffalo.edu (afternoon)
	→	clean up my mail folders (afternoon)
Likely to change plan?	→	yes
Details	→	web browsing

Table 1: An example Session-Scope

The watchdog will convert this into a sprint-plan which is a collection of verifiable assertions. A manually generated sprint-plan for the session-scope of Table 1 is illustrated in Table 2.

The subject field of the assertion statement is assigned on the basis of the knowledge that the workstation of user shambhu has the tty number as 10 and the lab machines at 600 Bell Hall have tty numbers as 12 and 120.

Now if a resolved user command is shambhu:tty25;rm, this command will not pass verification. Since it is a userID violation, it will be flagged as an intrusion.

5 Integrating Detection and Recovery

Most recovery and response generation techniques assume the presence of a mechanism for intrusion detection. In this document, we have presented the architecture and design of a concurrent intrusion detection scheme. The low latency characteristic of the detection mechanism can be exploited to provide a timely recovery and establish consistency across the network. Figure 3 below gives a simple architecture of the merged detection and recovery.

The individual watchdog processes on a given node of a distributed network monitor each user's commands and raise a exception when an intrusion occurs in a user account. Recovery

shambhu:tty10:tty12:tty120;latex;files;8am-5pm
shambhu:tty10:tty12:tty120;bibtex;files;8am-5pm
shambhu:tty10:tty12:tty120;spell;files;8am-5pm
shambhu:tty10:tty12:tty120;emacs;files;8am-5pm
shambhu:tty10:tty12:tty120;vi;files;8am-5pm
shambhu:tty10:tty12:tty120;lpq;queue;8am-5pm
shambhu:tty10:tty12:tty120;xdvi;files;8am-5pm
shambhu:tty10:tty12:tty120;ghostview;files;8am-5pm
shambhu:tty10:tty12:tty120;lpr;printers;8am-5pm
shambhu:tty10:tty12:tty120;cd;directories;8am-5pm
shambhu:tty10:tty12:tty120;pwd;directories;8am-5pm
shambhu:tty10:tty12:tty120;elm;folder;8am-5pm
shambhu:tty10:tty12:tty120;mail;folder;8am-5pm
shambhu:tty10:tty12:tty120;netscape;terminals;12pm-5pm
shambhu:tty10:tty12:tty120;cc;files;12pm-5pm
shambhu:tty10:tty12:tty120;gcc;files;12pm-5pm
shambhu:tty10:tty12:tty120;seqsim;files;12pm-5pm
shambhu:tty10:tty12:tty120;chmod;files;12pm-5pm
shambhu:tty10:tty12:tty120;rm;files;12pm-5pm
shambhu:tty10:tty12:tty120;mv;files;12pm-5pm

Table 2: A set of Verifiable Assertions

can be accomplished by periodically saving the user's system state and the transactions so that when an intrusion is detected, the user process is restarted from a previous state and the transactions are replayed. However, in a distributed setup, the recovery is much more complex. This is because, the user may have exchanged communication with other processes across the network and some of the data used by other processes may have been compromised. Thus, it is necessary to coordinate the recovery.

Figure 3 shows the scenario where the network watchdog performs intrusion detection based on the data it receives from the master watchdogs and the file watchdog on the various nodes. A signal is issued to a recovery process running on one of the nodes in the network. This recovery process then determines the extent of damage, terminates the suspected user(s) and initiates recovery. The recovery can take place by forcing all the processes in the network to rollback to their respective safe checkpoints and then recover as a community. Alternatively, the recovery process can determine which specific processes are affected and recover only those specific processes. This depends on the checkpointing strategy and is outside the scope of this document.

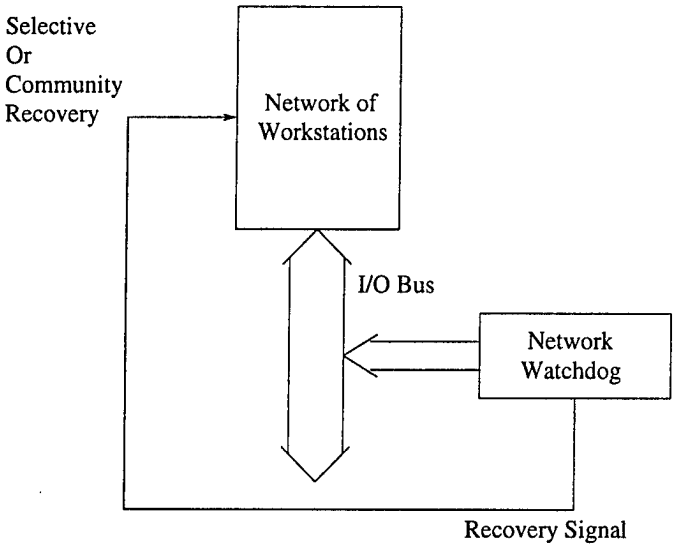


Figure 3: An Architecture for CID-Supported Recovery Scheme

6 Discussion

In this document, we have presented a new approach to intrusion detection using verifiable assertions. We have developed this scheme by leveraging some of the successful concepts from the fault tolerance literature. The main feature of our technique is that detection occurs concurrently with user operations and is real-time. The low latency detection of our technique can potentially speed up the recovery of affected systems. This is a significant benefit compared to the schemes based on audit trail analysis.

We have given a basic architecture and an algorithm for intrusion detection and illustrated the operation of the scheme. Several enhancements to the basic scheme are also presented. The technique is flexible in that changes or updates to the intended plan can be made easily. Also, every time a fresh session is started, a new set of verifiable assertions is generated. The finite length of a given user session helps to keep the sprint-plan to a small and bounded set.

The proposed technique is more effective in detection when intrusion occurs into logged on user accounts. This is because the intruder's operations are likely to result in large deviations from the intended session-scope of the user. Therefore, deactivating user accounts that are inactive for certain length of time will increase the probability of intrusion occurring in active user accounts where they can be detected. In the case where intrusion occurs when the (active) user is not logged on, the intruder routinely gets queried for a session-scope. This can be viewed as a deterrent because the idea that the system usage is monitored on-line may turn away 'casual' intruders.

6.1 Comparison to Related Work

The watchdog process which performs user session-level monitoring is the core module of the distributed concurrent intrusion detection scheme described in this document. Our scheme is not intended to be a replacement to other intrusion detection tools such as pattern matching and rule-based systems built to work on audit trail data. The concurrent monitoring watchdog described here can be used in conjunction with other distributed intrusion detection schemes to provide a higher detection resolution. For instance, the watchdog can be added as a third party security module in Emerald's monitor [11].

Emerald is a generic network monitor which uses audit data, network datagrams, SNMP traffic, application logs and analysis results from other intrusion-detection instrumentation. It uses a combination of statistical profile-based anomaly detection and signature-based inference mechanism.

There is some similarity between the watchdog and Emerald's monitor in that both provide a focused and distributed signature analysis. But the signature analysis of Emerald's monitor is essentially a rule-based expert system. On the other hand, the watchdog uses verifiable assertions that are generated on-line and compared with user's intended operations. This is also distinct from Anderson's program profiling [16].

The autonomous agents described in [12] are similar to Emerald's monitor and our watchdog. The transceivers correspond to the master watchdog of our technique and the entire monitoring is done in a distributed way. Similar to Emerald's monitor, the agents work on SNMP data and audit trail data. Unlike our scheme, it does not bind the agent with a user in a manner described in our scheme which allows for concurrent monitoring. Like ours, both monitor and agent are ongoing efforts and complete prototypes are not yet available.

6.2 Future Plan

Our current plan is to develop a graphical user interface for the easy user input of session-scope data. We also have plans to automate the generation of the assertions using formal techniques such as accounting checks or reasonableness checks [15].

There are several open problems. For instance, what exactly is a session? Is it a daily plan or a weekly plan or is it much shorter? If it is a daily plan, what are the start and end instances? This is an important issue because a user may dial-up from home and conduct work outside normal hours. The optimal length of a session in fact depends on the user behavior. This part needs further investigation. Establishment of proper tolerance limits in the algorithm to start a dialog with the user is also an important problem. There is a tradeoff between intrusion detection and false alarms that needs to be worked out. As a long term objective, we would like to put together a simulation engine to verify our concepts in a distributed computing environment. It is also desirable to actually implement the watchdog in a distributed system and perform real experiments.

References

[1] T. Lunt, "A survey of intrusion detection techniques," *Computers and Security*, vol. 12, pp. 405–418, 1993.

[2] J. Feldman, J. Giordano, and J. Palmer, "Information survivability at rome laboratory," *1997 IEEE Information Survivability Workshop*, 1997.

[3] M. Namjoo, "Techniques for concurrent testing of VLSI processor operation," *Proc. International Test Conference*, pp. 461–468, November 1982.

[4] P. Ammann, S. Jajodia, C. McCollum, and B. Blaustein, "Surviving information warfare attacks on databases," *1997 IEEE Symp. on Security & Privacy*, pp. 164–174, May 1997.

[5] D. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, pp. 222–232, February 1987.

[6] A. Clyde, "Insider threat identification systems," *Proc. 10th National Computer Security Conf.*, Sept. 1987.

[7] S. Kumar and E. Spafford, "A pattern matching model for misuse intrusion detection," *Proceedings of the 17th National Computer Security Conf.*, pp. 11–21, October 1994.

[8] K. Ilgun, R. Kemmerer, and P. Porras, "State transition analysis: A rule-based intrusion detection approach," *IEEE Trans. on Software Eng.*, vol. 21, pp. 181–199, March 1995.

[9] J. Hochberg, K. Jackson, C. Stallings, J. McClary, D. DuBois, and J. Ford, "NADIR: An automated system for detecting network intrusions and misuse," *Computers and Security*, vol. 12, pp. 253–248, May 1993.

[10] S. Snapp, S. Smaha, T. Grance, and D. Teal, "The DIDS Distributed intrusion detection system prototype," *USENIX, 1992 Technical Conference*, pp. 227–233, June 1992.

[11] P. Porras and P. Neumann, "EMERALD: Event monitoring enabling responses to anomalous live disturbances," *National Information Systems Security Conf.*, pp. 353–365, Oct. 1997.

[12] J. Balasubramaniyan, J. Omar, G. Fernandez, E. Spafford, and D. Zamboni, "An architecture for intrusion detection using autonomous agents," *Department of Computer Sciences, Purdue University, Coast TR 98-05*, 1998.

[13] M. Schutte and J. Shen, "Processor control flow monitoring using signatured instruction streams," *IEEE Transactions on Computers*, vol. C-36, pp. 264–276, March 1987.

[14] S. Upadhyaya and B. Ramamurthy, "Concurrent process monitoring with no reference signatures," *IEEE Transactions on Computers*, vol. 43, pp. 475–480, April 1994.

[15] D. Pradhan, *Fault tolerant computer system design.* Prentice-Hall, 1996.

[16] J. Anderson, "Computer security threat monitoring and surveillance," *Technical Report, James P. Anderson Co.*, April 1980.

CO-CHANNEL SPEECH AND SPEAKER IDENTIFICATION STUDY

Robert E. Yantorno
Associate Professor
Electrical & Computer Engineering Department

College of Engineering
Temple University
12th & Norris Streets
Philadelphia, Pa 19122-6077

Final Report for:
Summer Research Faculty Program
Rome Labs

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC

and

Speech Processing Lab
Rome Labs
Rome, New York

September 1998

CO-CHANNEL SPEECH AND SPEAKER IDENTIFICATION STUDY

Robert E. Yantorno
Associate Professor
Electrical & Computer Engineering Department
Temple University

Abstract

This study was comprised of two parts. The first was to determine the effectiveness of speaker identification under two different speaker identification degradation conditions, additive noise and speaker interference, using the LPC cepstral coefficient approach. The second part was to develop a method for determination of co-channel speech, i.e., speaker count, and to develop an effective method of either speech extraction or speech suppression to enhance the operation of speaker identification under co-channel conditions. The results of the first part of study indicate that under conditions of the same amount of either noise or corrupting speech, for example 0 dB SNR or TIR (target-to-interference ratio), noise is much more detrimental than corrupting speech to the operation of the speaker identification. For example, with 100% of 0 dB corrupting speech there still occurs a certain number of correct speaker identifications, i.e., about 40% accuracy. Ten (10) dB TIR interfering speech, as well as small amounts of interfering speech, i. e., 40% 0 dB TIR are not as detrimental to speaker identification. The results of the second part of the study indicate that a system for speaker count and speaker separation is possible. The harmonic sampling approach, developed during the study, uses the periodic structure of the fine structure of the frequency characteristics of voiced speech. Successful reconstruction of a single speaker indicates the potential of this approach as a candidate for speech separation. Also, it was shown that detection of co-channel speech is possible using the harmonic sampling approach. Further improvements as well as other possible approaches to the co-channel speech problem are discussed.

CO-CHANNEL SPEECH AND SPEAKER IDENTIFICATION STUDY

Robert E. Yantorno

Introduction

Co-channel speech is defined as a speech signal which is a combination of speech from two talkers. The goal of co-channel research has been to be able to extract the speech of one of the talkers from the co-channel speech. This can be achieved by either enhancing the target speech or suppressing the interfering speech. This co-channel situation has presented a challenge to speech researchers for the past 30 years. There are systems where the separation of two speakers is possible and this is well documented in the literature. However, this requires that there be more than one sensor (in the case of speech, more than one microphone) and therefore, by making use of the dissimilar recording conditions, the speech from two different speakers can be extracted, for example Chang *et al* (1998) and Weinstein *et al* (1993). Some recent investigations conducted on co-channel speaker separation are; Savic et al (1994), Morgan *et al* (1995), Benincasa & Savic (1997), Yen & Zhao (1997) for speech recognition and Meyer *et al* (1997) using the approach of amplitude modulation mapping and reassignment (a vector quantization process). However, separation of the target speaker from co-channel speech has been very difficult. Therefore, to make the problem more manageable, it is worthwhile to ask what is the final use of the target speech. For example, if the final goal is that a human listener will use the speech, then intelligibility and quality would be important characteristics of the extracted speech. However, if the extracted speech is to be used for speaker identification, then one would be concerned with how much and what type of target speech is need to perform "good" speaker identification, i.e., voiced and unvoiced speech or just voiced speech. Therefore, determining the effect of speaker interference on speaker identification would be of considerable interest. Also, the development of an effective target speaker extraction technique, which would provide for major improvement of co-channel speech, would also be a very useful tool. The goal of this study is to better understand how interfering speech degrades the functioning of speaker identification, and to also develop a method for extracting enough target speech from co-channel speech to provide sufficient speech information about the target speaker that one can make a reliable identification of the target speaker.

The situation with co-channel speech can be viewed in three different ways, i.e., as either an extraction of the target speech, as a suppression of the interference speech, or as an estimation of both speech signals. All these methods have been developed and each requires a very different approach. Therefore, study of the effect of speaker interference on speaker identification would be helpful in choosing between extraction, suppression or estimation, and would also provide information on the development of the method to assist in speaker identification under varying co-channel conditions.

Speaker Identification Study

As outlined above, the first part of the study was to determine the effectiveness of speech identification using the LPC cepstral coefficient approach under two different speaker identification degradation conditions, additive noise and speaker interference. Information on that part of the study is discussed below.

Additive Noise

Methodology

For the initial part of the study, speech material was taken from the Timit database. The number of speakers used for training was 15 males and 15 females. The number of files for training for each speaker was 5. The files were taken from the dialect region 1 (dr1 subdirectory). Also, 15 male and 15 female speakers were used for testing, and were taken from the same dr1 subdirectories. It should be noted that the test speakers were the same speakers used for training, i.e., the speaker identification tests were conducted under closed conditions. The files used for training were the "sx" prefix speech files and the testing files were "sa" and "si" prefix speech files, which were all different speakers speaking the same utterance.

Results

The initial part of section 1 of this study was to determine the effect of noise on the accuracy of the speaker identification. The amount of noise added was varied, either by adding a specific amount of noise, in dB, to the entire utterance or by adding a percentage of 0 dB noise to the speech. The range of added noise in dB was 0 to 30 dB, and the percentage range of 0 dB added noise was 20 to 100 percent. Noise was always added to the center of the utterance. This was

done to ensure that even at a low percentage of added noise, i.e., 20% noise, the probability of noise being placed in a region of speech would be greater than if the noise was placed at the beginning or end of the speech file. This was done because for most utterances there is a certain amount of silence prior to the onset of speech. Therefore, if noise was added to the beginning of the speech file, a certain part of the noise would always be added to silence, and would not contribute to the degradation of the speaker identification process. Results of the percent noise added experiments are shown in Figures 1a. and 1b. One important observation that can be made is that there is an almost linear inverse relationship between percent correct speaker identification and percent noise (figure 1b.).

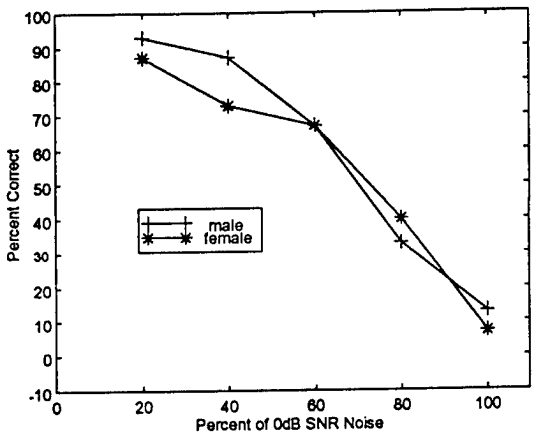


Figure 1a.

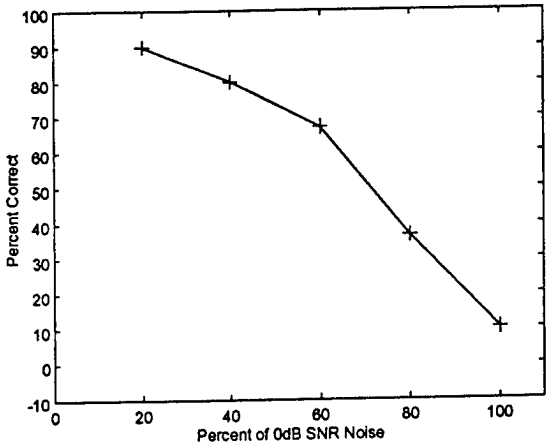


Figure 1b.

Figure 1. Speaker Identification – Percent Correct versus 0 dB SNR of Noise added to speech. Figure 1a. – male and female speakers. Figure 1b. represents the combined results of figure 1a.

The results of varying the signal-to-noise ratio are shown in figures 2a and 2b. The most dramatic decrease in speaker identification occurs between 20 and 10 dB. Also, it can be noted in figure 2b that 10 dB SNR is enough to totally degrade the speaker identification operation. Openshaw & Mason (1994) obtained similar results, where the effects of noise on the speaker identification using both the mel-cepstral coefficient technique as well as perceptual linear prediction-RASTA method were studied.

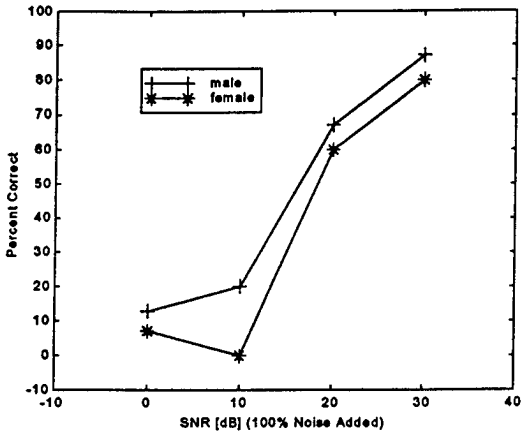


Figure 2a.

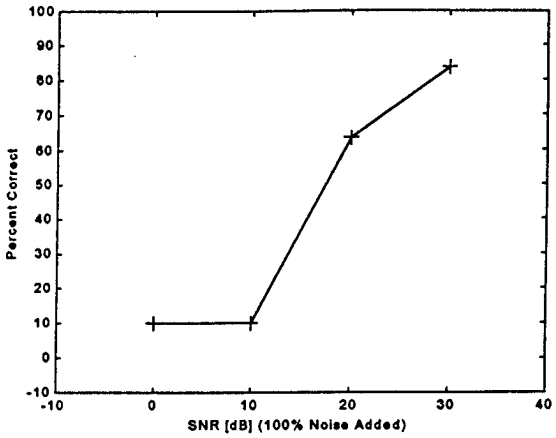


Figure 2b.

Figure 2. Speaker Identification – Percent Correct versus SNR of noise in dB added to speech (100% of speech corrupted by noise). Figure 2a. – male and female speakers. Figure 2b. is the combined result of figure 2a.

If one assumes that the noise experiments represent extreme conditions for speech corrupting speech, then certain conclusions can be drawn about the level, in dB, and amount of 0 dB speech that might be tolerable before one will observe a decease in percent correct speaker identification of co-channel speech. For example, one could tolerate about 40% of 0 dB SNR corruption speech with only a slight decrease of about 15 percent in percent correct (figure 1b.). Also, any corrupting speech of 20 dB target-to-interfere ratio (TIR), also referred to as signal-to-interference ratio (SIR), should have little effect on the speaker identification, in this case about 20 percent decrease in speaker identification percent correct. Therefore, the noise experiments provide a lower bound measure from which one can infer how well the speaker identification system will work under co-channel conditions.

Speaker Interference

Methodology

To determine the effect of corrupting speech on the accuracy of speaker identification, corrupting speech was added to test files. The amount of corrupting speech added was varied either by adding a specific amount of corrupting speech, in dB, to the entire utterance or by adding a percentage of 0 dB target-to-interfere ratio (TIR). The range of corrupting speech in dB was 0 to 30 dB TIR, and the percentage of 0 dB TIR ranged from 20 to 100 percent. As with the noise experiments, and for the same reason, corrupting speech was added to the center of the utterance.

Results

Two sets of four experiments were conducted. For one set, the corrupting speech was drawn from one of the speakers of the training and testing data, but was not the same utterance as used for that speaker's training or testing. These experiments are identified as "closed set" experiments and the results are shown in figures 3a. and 3b.

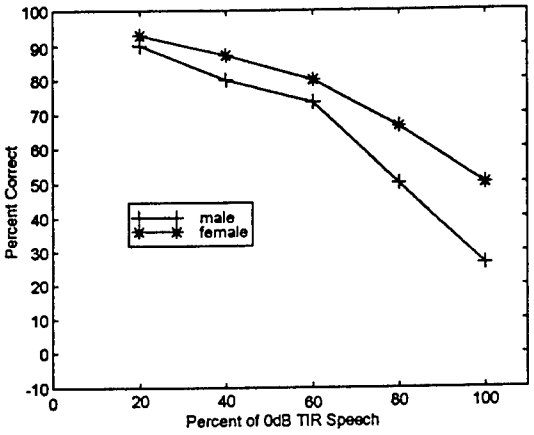


Figure 3a.

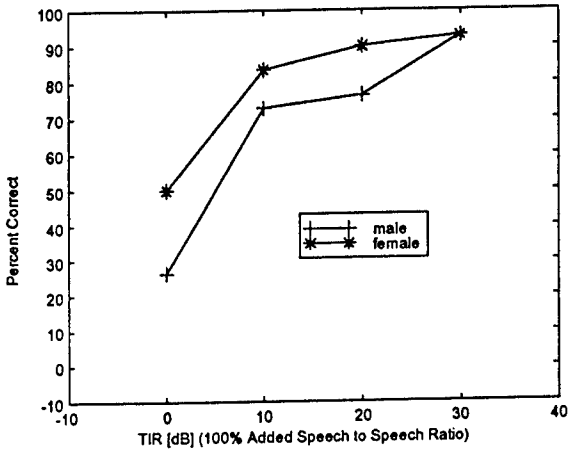


Figure 3b.

Figure 3. "Closed Set" Speaker Identification Experiments. Figure 3a. Percent Correct versus Percent of 0 dB TIR (Target-to-Interferer Ratio). Figure 3b. – Percent Correct versus TIR in dB of corrupting speech added to speech (100% of speech corrupted by speech).

For the other set of experiments, the corrupting speech was drawn from speakers outside the training and testing data. These experiments are identified as "open set", and the results are shown in figures 4a. and 4b. For both the closed and open set experiments there were four different types of experiments: 1.) male speech corrupted by either male or 2.) separately by female speech (results are identified as male in figures 3 and 4) and 3.) female speech corrupted by either male or 4.) separately by female speech (results are identified as female in figures 3 and 4).

One major observation that can be made with respect to figures 3 and 4 is that even with 100% corruption of 0 dB TIR there still exists a certain number of correct speaker identifications, i.e., about 40% accuracy. This indicates that corrupting speech has a smaller effect on the speaker

identification system than does noise, substantiating the point made earlier about noise contamination of speaker identification being the "worst" case.

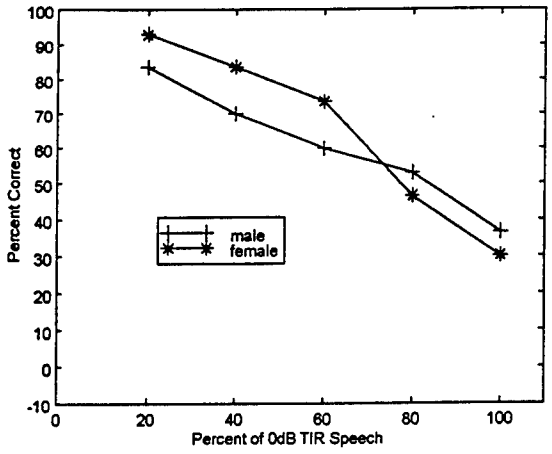


Figure 4a.

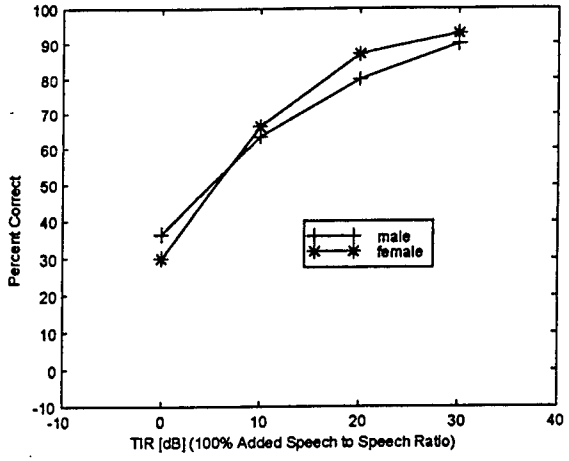


Figure 4b.

Figure 4. "Open Set" Speaker Identification Experiments. Figure 4a. Percent Correct versus Percent of 0 dB TIR. Figure 4b. – Percent Correct versus TIR in dB of corrupting speech added to speech (100% of speech corrupted by speech).

This result seems reasonable because 0 dB TIR does not spread the energy over the entire utterance as in the case of the noise experiments. It is also evident that although there is an almost linear inverse relationship between percent correct and percent added corrupting speech (figure 4a.), as in the case of noise (figure 1b.), the slope is not as steep, again as expected. Also, for the closed set experiments, figures 3a. and 3b., male speaker identification appears to be more sensitive to corrupting speech than does female speaker identification. A smaller effect can be observed with the open set experiments shown in figures 4a. and 4b.

Finally, a comparison is made between the "open" and "closed" set experiments and the results are shown in figures 5a. and 5b. The major observation to be made is that for speaker identification, corrupting speech with speech from outside the training data tends to have a greater effect on the percent correct than corrupting speech from within the training data, i.e., for the 100% of 0 dB TIR corruption speech condition the percent correct was 40% (for the closed condition) and 35% (for the open condition), or about 5% decrease in percent correct.

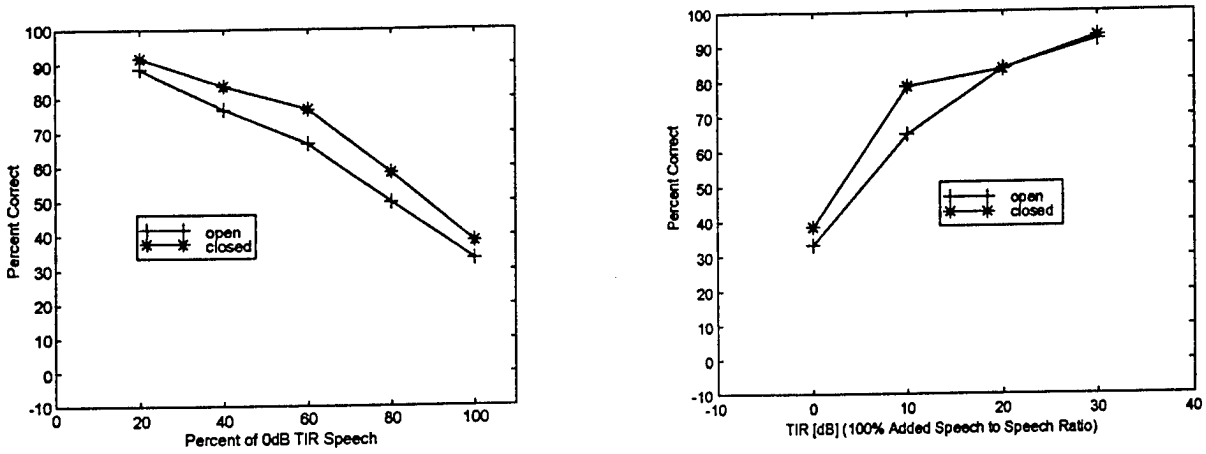


Figure 5. Comparison of data from "Closed Set" and "Open Set" Speaker Identification Experiments. Figure 5a. Percent Correct versus Percent of 0 dB TIR. Figure 5b. – Percent Correct versus TIR in dB of corrupting speech added to speech (100% of speech corrupted by speech).

It should be noted that Yu & Gish (1993) obtained comparable results for experiments similar to the ones shown in figures 3 and 4. Their goal was to identify either one or both speakers engaged in dialog using speech segments rather than frames and speaker clustering.

Co-channel Speech

Introduction

The second part of this project was to develop a method for effective determination of co-channel speech, i.e., speaker count, and be able to extract enough speech information about the target speech so that a reliable identification of that speaker could be made. The method which was developed is based on harmonic sampling in the frequency domain, and is outlined below. It should be noted that the method presented here is similar to the approaches of Doval & Rodet (1991) and Casajus Quiros & Enriquez (1994), which were used for fundamental frequency determination and tracking of music signals. It should also be mentioned that there is some similarity between the method outlined here and the maximum likelihood pitch estimation developed by Wise *et al* (1976). However, Wise *et al* used the autocorrelation and the time domain for their approach, whereas the frequency domain and the magnitude spectrum will be used for the method outlined here. However, they did mathematically analyze their method in the

frequency domain and determined that maximizing their peak estimator was equivalent to finding the comb filter which passes the maximum signal energy, which is the basis for the harmonic sampling method presented in this study.

Harmonic Sampling Method

If one observes the frequency domain characteristics of a voiced portion of a speech signal, it can be noted that there are two distinct attributes, the spectral envelope of the speech signal, and the fine structure which is a series of pulses. The spectral envelope consists of the frequency characteristics of the vocal tract. The fine structure consists of the frequency characteristics of the excitation which is the input to the vocal tract. The excitation for voiced speech is characterized by periodic time pulses produced by the vocal cords which produce periodicity in the frequency domain. The fine structure and its periodicity are illustrated in figure 6 below. The periodicity of the fine structure is the basis for the approach presented. Because we are using the fine structure, and the fine structure only exists during voiced speech, this means that only the voiced portions of speech will be used. Also, voiced speech appears to carry much more speaker identification information than unvoiced speech.

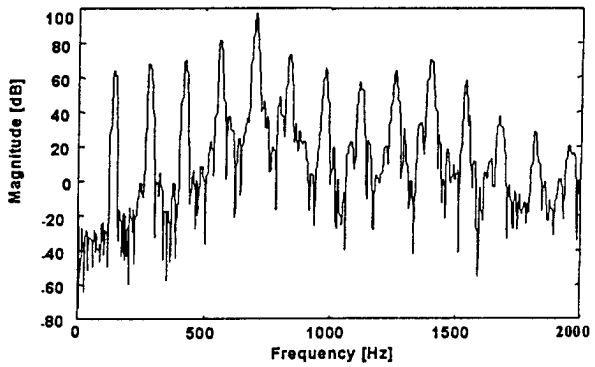


Figure 6. Frequency characteristics of a frame of speech – magnitude in dB versus frequency in Hz, 800 point frame, Hamming windowed, and sampled at 8 kHz.

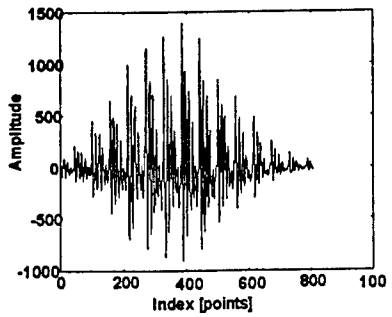


Figure 7a.

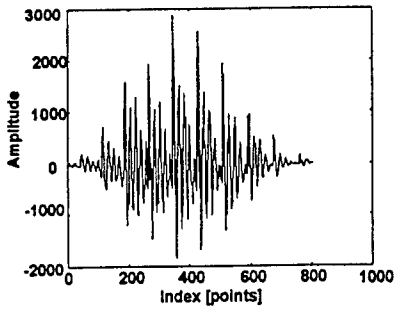


Figure 7b.

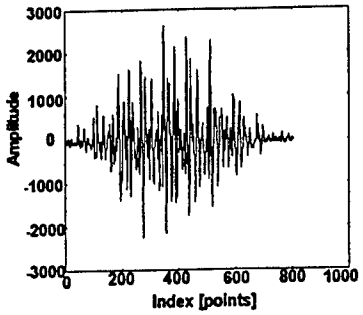


Figure 7c.

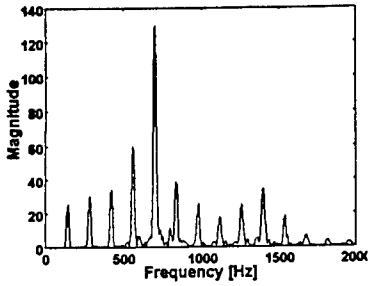


Figure 7d.

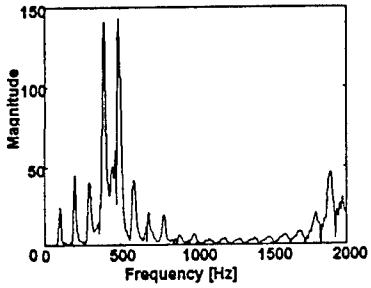


Figure 7e.

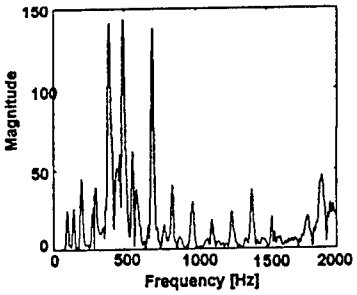


Figure 7f.

Figure 7. Time and frequency characteristics of speech from single speakers - speaker #1 (7a. and d.), speaker #2, (7b. and 7e.) and combined speakers #1 and #2 (7c. and 7f.)

For the case of co-channel speech, it is expected that the frequency characteristics of two speakers is additive. However, the magnitude characteristics are additive but also contain a cross term. Therefore, the overall fine structure will be a quasi-linear combination of fine structure of the two speakers' speech. This is illustrated in figure 7f. As stated, the approach in this study uses the fine structure as a method for determining if there is more than one speaker present and also, very importantly, uses the information obtained as a means of extracting the target speech.

The experiment, for speaker count and speaker separation, entailed using a variable spacing inverse comb filter to sample the magnitude spectrum. If the spacing between the filter lines and between the first line and the vertical axis are variable then one has a tunable comb filter. Therefore, there should be a maximum when the spacing between the comb lines is equal to the fundamental frequency of the speech, as discussed previously and stated by Wise *et al* (1976) for

their pitch detection method. Using the harmonic sampling method, the frequency spectrum is "swept", sampling the spectrum at discrete frequencies, and adding all of samples, in this case 31, at each frequency step. The result will be a graph with a peak at the fundamental frequency. However, after some work on filter design and further considerations, it was recognized that one need only sample the spectrum, and therefore, the comb filter was not needed. A series of diagrams of harmonic sampling for various values of sampling spacing are shown in figure 8 below.

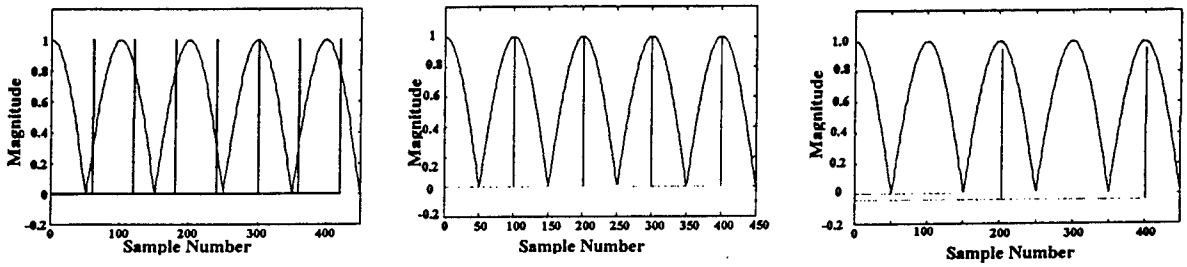


Figure 8a. Figure 8b. Figure 8c.

Figure 8. Harmonic sampling. Three different harmonic sampling spacings, less than fundamental (8a), at fundamental (8b), and at 2x fundamental (8c).

The sampling of the magnitude spectrum results in a peak at the fundamental, but also results in "harmonic-like peaks" at locations related to the fundamental. This situation is illustrated in figure 8c. It is evident that there will be a peak due to the spacing as illustrated in figure 8c. Because we are using a fixed number of lines, the height of that peak will be about half the height of the main peak. However, there will also be a peak at half the fundamental, for example, using figure 8b., if there were twice as many lines as shown, there would be lines located halfway between the lines shown, at the nulls of the spectrum.

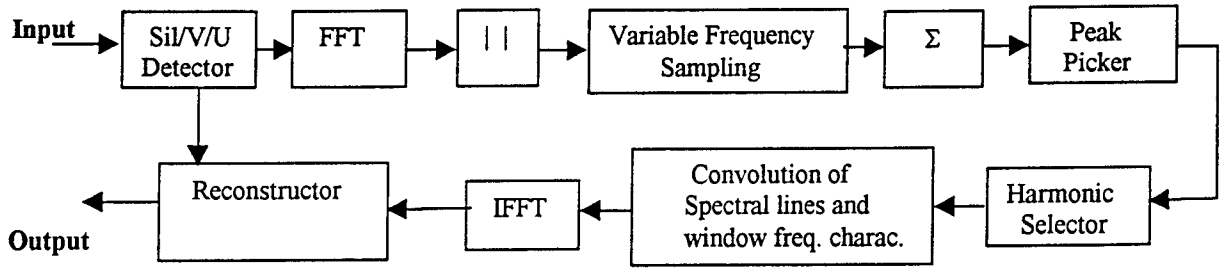


Figure 9. Block diagram of target extraction procedure. Input is co-channel speech and output is target speech.

The height of this peak will also be about one-half the height of the main peak. The entire harmonic sampling method is outlined in the block diagram, shown in figure 9 above.

A peak-picking algorithm was developed to determine the main peak associated with the stronger speaker's speech. To illustrate the effectiveness of the peak-picking algorithm uncorrupted speech was used. The spectrum of uncorrupted speech was sampled at the fundamental and all of the harmonics up to harmonic 30. A plot of the magnitude spectrum, the sampled harmonic spectrum, and the reconstructed magnitude spectrum are shown in figures 10a., 10b. and 10c. respectively.

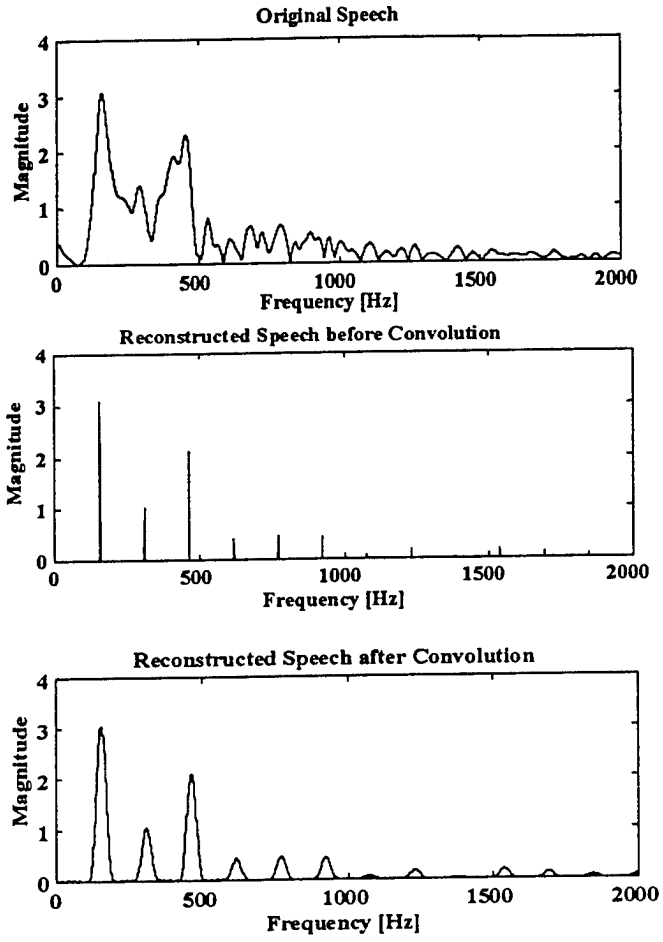


Figure 10. Magnitude frequency plots for original (upper), harmonically sampled (middle) and reconstructed spectrum (lower).

It was determined that the magnitude spectrum provided a much better main peak, in terms of height above the other random peaks, than either the power spectrum or log magnitude spectrum. Also, a slight positive slope was observed for the harmonic sampling results. A fourth order polynomial fit of the data was subtracted from the data to provide a better environment for peak picking. Windows of various lengths were investigated, and a length of 400 points appeared to be the optimal in terms of frequency-time resolution and shape and size of the major harmonic peak. The 400-point frame was windowed using a Hamming window and was zero padded to 8000 points prior to performing the FFT. This results in each point in the harmonic sampling result plot to be equal to 1 Hz. It should be noted that this is not the resolution of the harmonic sampling approach. Note, the Timit speech data was down sampled prior to processing from 16 KHz to 8 KHz using Entropic's ESPS sfconvert utility. Fifty percent overlap was used during the analysis phase to compensate for the windowing of the frame.

Once the spectral lines are obtained, the convolution of these lines with the frequency characteristics for the Hamming window is necessary in order to obtain a "window" time function similar to the original speech frame from which the spectral data was obtained (see figure 10 above). Finally, for reconstruction, the frames were overlapped by 50% to duplicate the overlap process used for extracting and analyzing the speech frame. Using both the harmonically sampled magnitude and phase characteristics for reconstruction did not provide a very good reproduction of the new frame. Therefore, the entire phase characteristics were used for reconstruction. This results in very good duplication as shown in figure 12 below.

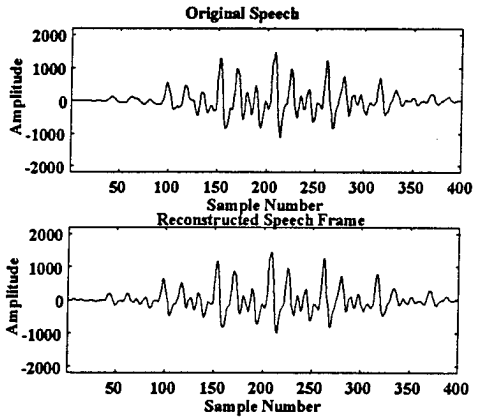


Figure 11. Windowed frame of speech, original speech (upper) and reconstructed speech (lower).

How well does this approach track the pitch of a single speaker? A voiced/unvoiced detector was obtained from Dan Benincasa and Stan Wenndt (of the Speech Processing Lab). The result, for male speech, is shown in the figure 12 below. It is evident that the algorithm works very well, and therefore this is a good candidate for use as a pitch tracker as well as a tool for speech extraction.

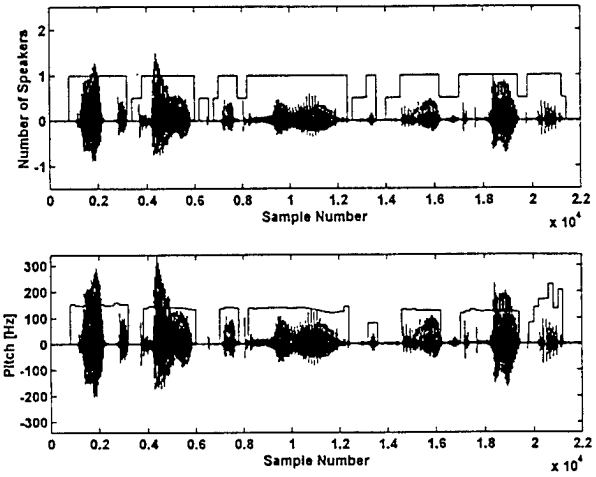


Figure 12a. (upper) and 12b. (lower)

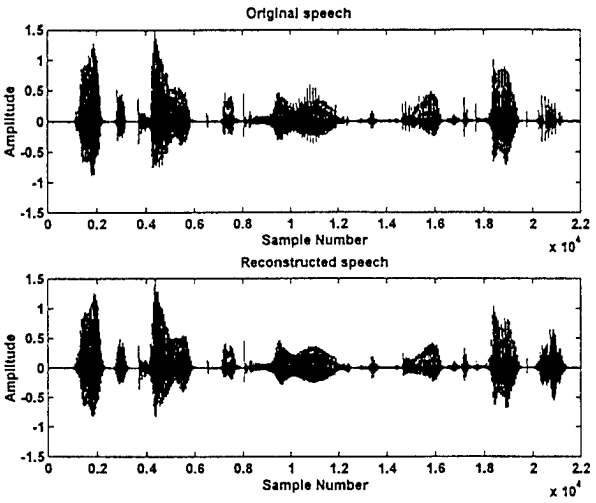


Figure 12c. (upper) and 12d. (lower)

Figure 12. Time plots of speech signals. Figure 12a. Identification of voiced (1), unvoiced (0.5) and silence (0) sections of speech signal. Figure 12b. Pitch versus time for voiced portions of utterance. Figures 12c. & 12d. Original and reconstructed speech, respectively.

How effective is the algorithm in determining the existence of two speakers? The results shown in figures 13a., 13b. and 13c. represent the results of speech data shown in figures 7a., 7b., and 7c., respectively for single speakers (figures 13a. and 13b.) and co-channel speech (figure 13c.). As can be noted, the pitch of both speakers is clearly seen in figure 13c., as marked by the straight lines in the middle of the two tallest peaks. Note the peak on the right in figure 13c is not at the location where one would expect a multiple of the pitch of the largest peak, and therefore the peak on the right represents the pitch of another speaker.

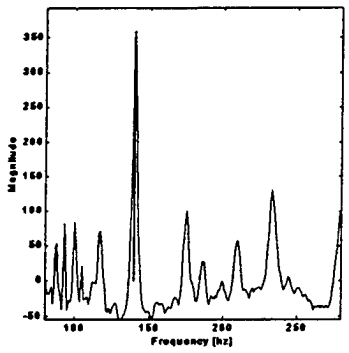


Figure 13a.

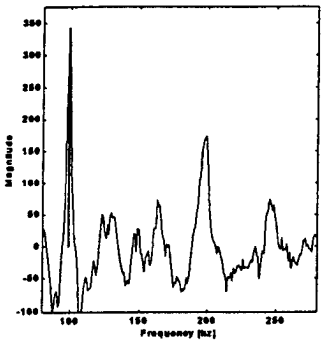


Figure 13b.

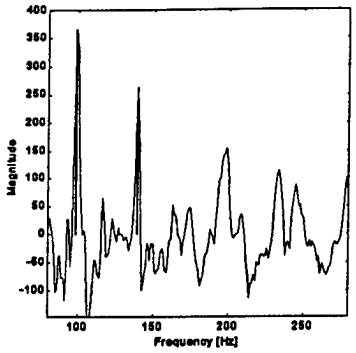


Figure 13c.

Figure 13. Harmonic sampling results for single speakers (speaker #1) figure 13a., (speaker #2) figure 13b., and for co-channel speech (a mixture of speaker #1 and speaker #2) figure 13c.

Conclusions

It has been shown that the harmonic sample approach can be used for speaker count, where the pitch of both speakers are not the same. It has also been shown that the harmonic sampling approach can be used to extract very effectively and quite accurately the voiced portions of a single speaker. The harmonic sampling approach outlined above offers promise as a mechanism for extracting the voiced portion of target speech. However, there are problems to overcome and improvements which can be made.

Further Improvements for the Harmonic Sampling Method

For example, the occurrence of small spurious peaks close to the main pitch peak need to be eliminated, possibly by smoothing the harmonic sampling resultant or by using their occurrence as an indicator not to use that frame of speech. Also, large spurious peaks far from the main peak need to be understood in terms their origin, and to be reduced, eliminated or possibly used as an indicator not to use that frame of speech. Also, the harmonic sampling method may be able to be used to determine if the frame being analyzed is voiced, unvoiced or silence, possibly by using some sort of indicator of the existence of large peaks for voiced, small peaks for unvoiced and no peaks for silence. The final item for investigation is to determine a way to identify speaker #1 speech and speaker #2 speech, and to track their speech. This will require more sophisticated ways in which the speech of a specific speaker is identified. One possibility would

be to use a pitch tracking approach such as the one developed by Doval & Rodet (1993) who used a probabilistic Hidden Markov model or model space, or the approach of Dorken & Nawab (1994) which uses principle decomposition analysis.

It should be noted that the approach outlined here might be useful as a pitch detector as well as a voiced/unvoiced detector. The method of Wise *et al* (1976) has been shown to be resistant to noise. Therefore, because of the similarity between harmonic sampling and maximum likelihood pitch estimation, harmonic sampling should also be resistant to noise.

Finally, this speech separation approach shows promise in terms of being able to extract the interfering speech by subtracting the reconstructed speech from the original speech using either the frequency domain or the time domain.

Areas of Possible Further Study

Speaker Count

I feel that developing a system for identifying co-channel speech is possible. However, to be able to identify co-channel speech will require using unorthodox types of approaches. One such approach would be to use the approach of speech recognition similar to that used for language identification. It seems reasonable that co-channel speech, which is the result of speech corrupted by speech, will not have the same time domain structure as traditional speech, and therefore will not have the same type of phonetic structure as single speaker speech. This means that recognition would not be successful and would therefore indicate the existence of co-channel speech.

It also seems possible that an effective speaker count approach could be developed using information in the time domain. It is evident from inspection of the co-channel speech that there is a dramatic change in the overall structure as compared with single speaker speech (compare figure 7c. with either figure 7b. or 7a.). If a time domain speaker count system could be made then this speaker count system could be used as the front-end of a speech separation system. This would ascertain whether a frame of speech is from a single or multiple speakers. If it is co-

channel speech then it would be processed by the speaker separation system. However, if it is from a single speaker the frame would not be processed, thereby reducing computation time as well as eliminating any possible degradation of the speech using the speech separation system.

Another possible approach would be to use LPC to determine the presence of co-channel speech. For example, if two speakers were talking at the same time, they usually would not be saying the same thing. Therefore, each speaker would be producing different speech sounds and each speaker would then have a different vocal tract configuration at any instant in time. This would seem to suggest that a series of LPC analyses could be done on a single frame. Assuming that one has co-channel speech, LPC analysis would be performed on the speech. Once a set of LPC coefficients had been obtained their effect could be subtracted from the co-channel speech by inverse filtering. Then performing a subsequent LPC analysis on the inverse filtered signal should produce another set of LPC coefficients only if co-channel speech is present. Note, this approach could only be used to detect co-channel speech. It would not be able to extract the target speaker's speech.

Speaker Separation

Although there is no information available about using singular value decomposition (SVD) for co-channel speech, it might be a possible approach. Kanjilal & Palit (1994) have used SVD as a means of extracting two periodic stationary waveforms from noisy data; in this case the waveforms were maternal and fetal electrocardiograms. It should be noted that their approach had no requirement for multiple sensors, but did require that the signals be stationary.

Finally, AM mapping and spectrum reassignment by Meyer *et al* (1997) seems to provide some promise as a means of separating speakers under co-channel speech conditions. They suggest that the modulation maps are good models for human perceptual data, and by using a reassigned spectral approach, frequency resolution is increased.

References

1. Benincasa, D. S. and Savic, M. I., "Co-channel speaker separation using constrained nonlinear optimization," Proc. IEEE ICASSP, pp:1195-1198, 1997.

2. Casajus Quiros, F. J. and Enriquez, P. F-C., "Real-time, loose-harmonic matching fundamental frequency estimation for musical signals," Proc. IEEE ICASSP, Ii-221-II-224, 1994.

3. Chang, C., Ding, Z., Yau, S. F. and Chan, F. H. Y., "A matrix-pencil approach to blind separation of non-white sources in white noise," IEEE ICASSP, pp: IV-2485-IV-248, 1998.

4. Doval, B. and Rodet, X., "Estimation of fundamental frequency of musical sound signals," Proc. IEEE ICASSP, pp:3657-3660, 1991.

5. Doval, B. and Rodet, X., "Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and HMMs," Proc. IEEE ICASSP, pp:I-221-I-224, 1993.

6. Dorken, E. and Nawab, S. H., "Improved musical pitch tracking using principal decomposition analysis," Proc. IEEE ICASSP, pp:II-217-II-210, 1994.

7. Kanjilal, P. P. and Palit, S., "Extraction of multiple periodic waveforms from noisy data," Proc. IEEE ICASSP, pp:II-361-II-364, 1994.

8. Meyer, G. F., Plante, F., and Bethommier, " Segregation of concurrent speech with the reassignment spectrum," Proc. IEEE ICASSP, pp:1203-1206, 1997.

9. Morgan, D. P., George, E. B., Lee, L. T, and Kay, S. M., " Co-channel speaker separation," Proc. IEEE ICASSP, pp:828-831, 1995.

10. Openshaw, J. P. and Mason, J. S., "On the limitations of cepstral features in noise,", Proc. IEEE ICASSP, pp: II-49-II-52, 1994.

11. Savic, M., Gao, H. and Sorensen, J. S., "Co-channel speaker separation based on maximum-likelihood deconvolution,", IEEE ICASSP, pp:I-25-I-28, 1994.

12. Weinstein, E., Feder, M., and Oppenheim, A. V., "Multi-Channel Signal Separation by Decorrelation," IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. 1, No. 4, pp:405-413, Oct. 1993.

13. Wise, J. D., Caprio, J. R., and Parks, T. W., "Maximum likelihood pitch estimation," IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-24, No. 5, pp:418-423, Oct. 1976.

14. Yen, K-C and Zhao, Y., "Co-channel speech separation for robust automatic speech recognition: stability and efficiency," Proc. IEEE ICASSP, pp:859-862, 1997.

15. Yu, G., and Gish, H., "Identification of speakers engaged in dialog," Proc. IEEE ICASSP, pp:II-383 – II-386, 1993.